

IWA (Ideal Workspace Agent)

Soporte para la evolución colaborativa de Configuraciones de Trabajo

Trabajo de Grado

Licenciatura en Informática
Facultad de Informática
Universidad Nacional de La Plata

Clasificación

Investigación Aplicada

Alumnas

María Paula Fernández
Nro. de Alumno: 1528/6
Teléfono: 453-8923
e-mail: mpaulafer@yahoo.com.ar

Valeria Améndola
Nro. de Alumno: 1479/5
Teléfono: 482-4974
e-mail: amendolava@yahoo.com.ar

Director

Lic. Luis Mariano Bibbó
LIFIA. Universidad Nacional de la Plata. La Plata. Buenos Aires. Argentina.

Codirector

Lic. Alejandro Fernández
LIFIA. Universidad Nacional de la Plata. La Plata. Buenos Aires. Argentina.

CONTENIDO

<i>CAPITULO 1 - Introducción.....</i>	<i>1</i>
1.1 Motivación	2
1.2 Objetivo.....	3
1.3 Organización de la Tesis.....	4
<i>CAPITULO 2 - Trabajo Colaborativo y Groupware</i>	<i>5</i>
2.1 Motivación de Aplicaciones Colaborativas.....	7
2.2 CSCW.....	8
2.3 Diferencias entre Groupware y CSCW.....	9
2.4 Comunicación, Colaboración y Cooperación	10
2.4.1 Cooperación y Colaboración en CSCW.....	11
<i>CAPITULO 3 - Tailoring.....</i>	<i>12</i>
3.1 Soluciones adaptables	12
3.1.1 Adaptabilidad como una característica crucial de groupware....	13
3.1.2 Contribución de los Componentes de Groupware a la Adaptabilidad	15
3.2 Contribución de la Adaptación al Aprendizaje.....	16
3.2.1 Aprendizaje Individual y Aprendizaje Colaborativo	18
<i>CAPITULO 4 - Evolución Colaborativa de Configuraciones de Trabajo.....</i>	<i>21</i>
4.1 Ambientes de Trabajo	21
Ejemplo: Taller mecánico	22
Paralelismo con un ambiente de trabajo informático.....	22
Evolución, Adaptación y Colaboración	23
4.2 Introducción al IWA.....	24
4.2.1 Ideal Workspace Agent.....	25
4.3 Conceptos Involucrados	26
4.4 Los Roles y el Entorno de Trabajo	28
4.4.1 Manejo de Roles.....	30
4.5 Evolución Colaborativa de una Configuración de Trabajo.....	30
Incorporación de un recurso a una configuración de trabajo	30
Eliminación de un recurso de una configuración de trabajo	31

4.6	Debate.....	33
4.7	Ejemplo	34
CAPITULO 5 - Ideal Workspace Agent: Implementación.....		36
5.1	Estructura de la Aplicación.....	36
	Integración con el Sistema Operativo	36
	Soporte de colaboración.....	36
	Soporte de Comunicación	37
	Estructura de la aplicación	38
5.2	Modelo Genérico de una Configuración de Trabajo	38
5.3	Extensión del Modelo Genérico para el caso elegido	40
5.4	IWA	42
5.5	Debate.....	44
5.5.1	Decisiones tomadas	44
	Alternativa 1: Discusión usando una modificación de Snapchat.....	44
	Alternativa 2: Discusión sin usar Snapchat.....	47
	Alternativa implementada	49
5.5.2	Modelo	50
5.6	Configuración Local	52
CAPITULO 6 - DyCE		55
6.1	Objetivos y Requerimientos	55
6.2	Diseño del Sistema.....	56
6.2.1	Componentes Groupware.....	57
6.2.2	Arquitectura del Sistema	57
6.2.3	Objetos Compartidos.....	59
6.2.4	Movilidad de Componentes (Component Mobility).....	62
6.2.5	Integración Web	62
6.3	Modelo de Programación de Componentes	63
6.3.1	Aplicaciones y Sesiones.....	65
6.3.2	Sesiones y Acoplamiento	66
6.4	¿Por qué DyCE?	66
6.5	Chatblocks	67
CAPITULO 7 - Conclusiones y Trabajo Futuro		69
7.1	Experiencia de uso	69
	Sistema Operativo.	69
	Java 69	
	Interacción con el Sistema Operativo	69
7.2	Trabajo Futuro.....	70
	Actualización bidireccional entre el IWA y el Sistema Operativo	70
	Extensión a otras plataformas y aplicaciones.....	71
7.3	Estado del Arte	71

7.4 Conclusiones	72
<i>ANEXO A - Trabajo Colaborativo y Groupware</i>	<i>73</i>
A.1 Taxonomías de Sistemas de Groupware	73
A.1.1 Taxonomía de Tiempo y Espacio.....	73
A.1.2 Taxonomía de Nivel de Aplicación.....	74
A.1.3 Taxonomía de Nivel de Compartición de Objetos.....	76
A.2 Perspectivas	77
A.3 Características de Sistemas Colaborativos	79
A.4 Aspectos de Diseño	81
A.4.1 Interfaces grupales	82
A.4.2 Procesos de grupos.....	83
A.4.3 Control de concurrencia	84
A.4.4 Otros aspectos de sistemas	85
A.5 Herramientas de groupware: como se caracterizan	86
A.5.1 GroupKit	86
A.5.2 Lotus Notes	87
A.5.3 Habanero	89
A.6 Ejemplos de aplicaciones Colaborativas	91
A.6.1 Colaboración basada en Sesiones (Talk++)	92
A.6.2 Colaboración basada en Mensajes (Mail++).....	95
A.6.3 Colaboración basada en Objetos (File++).....	96
A.6.4 Colaboración basada en Sesiones y Mensajes (Talk + Mail).....	98
A.6.5 Colaboración basada en Sesiones y Objetos (Talk + File).....	99
A.6.6 Colaboración basada en Mensajes y Objetos (Mail + File)	104
A.6.7 Colaboración basada en Sesiones, Mensajes y Objetos (File + Mail + Talk)	108
<i>Bibliografía.....</i>	<i>110</i>

CAPITULO 1 - INTRODUCCIÓN

Es común ver como los seres humanos tienden a adaptar los ambientes que frecuentan de acuerdo a sus necesidades. Por ejemplo, un mecánico pone aquellas herramientas que usa más a menudo en lugares fáciles de acceder y guarda en lugares más apartados las menos usadas. Este proceso de organización de los objetos que nos rodean, es casi inconsciente y tiene como fin agilizar el trabajo diario.

Actualmente, la mayoría de las empresas o equipos de trabajo utilizan la informática como medio para realizar sus tareas. Independientemente de la actividad que realicen o el área en que se desempeñen, la asistencia de una o varias computadoras les permite llevar a cabo sus actividades de manera más ágil, rápida y eficiente.

Podemos hacer una similitud entre el taller del mecánico y una computadora, a menudo los usuarios de estas últimas utilizan las mismas aplicaciones y/o documentos para realizar sus tareas cotidianas. De esta manera, observamos que dichas personas tienden a configurar su entorno de trabajo con las aplicaciones y documentos que usan más frecuentemente. Esto se percibe, por ejemplo, al observar como los usuarios de entornos X-Windows adaptan sus escritorios y menús de acceso rápido para las tareas que comúnmente realizan.

En la actualidad, el esquema más avanzado para modelar el entorno de trabajo está basado en identidades. Una configuración está generalmente determinada por una configuración básica a la que se le agrega la configuración dependiente de la identidad del usuario. Esta va evolucionando a medida que el usuario ataca nuevas tareas o desempeña distintos roles.

Dentro de un grupo de trabajo, se observa como los usuarios que realizan las mismas tareas tienden a utilizar las mismas herramientas o, por lo menos coinciden en el uso de muchas de ellas. Cada usuario, por su cuenta, consume parte de su tiempo, esfuerzo y energía en obtener un entorno de trabajo adaptado a sus necesidades. Al cabo de un tiempo, vemos que dentro de un mismo grupo existen varios entornos de trabajo similares, como resultado del esfuerzo de adaptación independiente de cada integrante. Es evidente la existencia de duplicación no intencionada de esfuerzos.

Las nuevas tecnologías informáticas ofrecen una gran variedad de aplicaciones destinadas a facilitar el trabajo grupal, brindando al usuario un nuevo y mejor espacio de comunicación y cooperación entre los miembros de su grupo. A estas tecnologías se las encuadra en el rubro de CSCW (Computer Supported Collaborative Work), es decir

el área del groupware, que emplea la computadora como el elemento básico sobre el que se apoyan las herramientas para el trabajo en grupo. Es de esperar que estas tecnologías sirvan para posibilitar la colaboración en la adaptación de los entornos de trabajo de usuarios con necesidades similares. Así, se podría obtener el entorno de trabajo deseado, reduciendo notablemente el tiempo y esfuerzo aportado por cada integrante, consiguiendo además, un aumento de la productividad del grupo.

1.1 MOTIVACIÓN

Como resultado del estudio del comportamiento de los grupos de trabajo, descrito en la sección anterior, surgió la idea de construir una herramienta colaborativa que permita a un grupo de personas que desarrollan las mismas tareas, configurar conjunta y evolutivamente sus entornos de trabajo. A esta herramienta la llamamos Ideal Workspace Agent (IWA).

La idea es, por medio de la asociación de recursos a roles e identidades, obtener de manera colaborativa un ambiente de trabajo “ideal”, adaptable y configurado de acuerdo a las actividades que desempeña cada usuario dentro de un grupo; utilizando como medio principal la discusión o negociación.

Detallando un poco mas la propuesta, lo que planteamos es que cada una de las personas tenga asociada la configuración del entorno de trabajo correspondiente a su identidad y a la del o los roles que esté desempeñando. A medida que estas progresan en sus tareas, les podría surgir la necesidad de utilizar nuevas y distintas herramientas. Ante estas situaciones, los usuarios que estén desempeñando un mismo rol, debatirían sobre la incorporación o no de la o las herramientas a la configuración de trabajo en cuestión. De esta forma, las configuraciones de trabajo evolucionarían de manera conjunta.

Lo que resulta interesante es que la posibilidad de configurar el entorno de trabajo en base a los distintos roles o actividades que cada integrante desempeña, permite a cada uno compartir su configuración con el resto de los usuarios que estén desarrollando su misma actividad, logrando que las personas trabajen de una manera más integrada, intercambiando experiencias personales, aprendiendo unas de otras y optimizando los tiempos de aprendizajes de los usuarios más novatos. Al mismo tiempo, con la forma de interacción propuesta, negociación y discusión, se obtendría una comunicación que explotaría los aspectos de la colaboración, incrementando y mejorando notablemente las relaciones humanas que se han visto desfavorecidas con las formas de trabajo donde cada persona trabaja con su computadora de forma aislada e independiente.

1.2 OBJETIVO

Este trabajo tiene como objetivo principal obtener una herramienta como la descrita anteriormente, de uso sencillo y práctico, que se adapte a las necesidades de los grupos de trabajo asistidos por computadora, permitiendo que cada uno de los distintos perfiles que participen en un grupo, obtenga colaborativamente una configuración “ideal” de sus entornos de trabajo. Para demostrar la importancia y las ventajas que se pueden obtener al utilizar las tecnologías basadas en el concepto de groupware.

La implementación de la herramienta de ejemplo está basada en un framework denominado DyCE que esta totalmente construido y basado en la tecnología Java.

Nuestra decisión de implementar el IWA como un componente del DyCE, se basa en que este framework ataca una serie de puntos a tener en cuenta en el desarrollo de dichos componentes:

- Permite trabajar con objetos compartidos.
- Asegura la consistencia de los datos compartidos.
- Implementa un mecanismo de notificaciones para informar de los cambios que se produzcan en los objetos compartidos.
- Emplea un paradigma de programación que hace flexible y extensible la incorporación de nuevos componentes groupware al sistema.

Para la parte de discusión o negociación usamos otro framework de caja blanca denominado Chatblocks, que está desarrollado sobre DyCE. Este permite la construcción de herramientas de comunicación sincrónica basada especialmente en texto, conocidas dentro de los ambientes groupware como Sistemas “Chat”.

Podemos resumir los objetivos de este trabajo en las siguientes tareas:

- Concebir un modelo que permita modelar el concepto de entorno de trabajo de forma que pueda evolucionar de acuerdo a las necesidades de un grupo de trabajo.
- Desarrollar un esquema de implementación independiente de la plataforma de software para el soporte de entornos de trabajo "evolucionables".
- Especializar el modelo anterior para el caso de un entorno Microsoft Windows.
- Realizar pruebas piloto con el objetivo de evaluar los resultados del uso de dicho enfoque.
-

1.3 ORGANIZACIÓN DE LA TESIS

El objetivo de esta sección es describir como esta organizado este trabajo y presentar los temas que abarca.

Comenzamos describiendo el Trabajo Colaborativo y las herramientas de Groupware en el capítulo 2. Presentamos más detalle de este tema en el Anexo A.

En el capítulo 3 introducimos el concepto de Tailoring y como los usuarios adaptan sus herramientas de acuerdo a sus necesidades.

En el capítulo 4 analizamos las Configuraciones de Trabajo actuales y como podría mejorarse su uso, también describimos como estas configuraciones podrían evolucionar colaborativamente. Este capítulo, junto al siguiente, es uno de los más importantes porque es dónde presentamos la solución propuesta.

En el capítulo 5 contamos en detalle como implementamos la herramienta: Ideal Workspace Agent.

En el capítulo 6 presentamos el framework DyCE, necesario para poder entender nuestro trabajo práctico, debido a que el mismo está basado en dicho framework.

En el último capítulo, se encuentran las conclusiones y trabajo futuro.

CAPITULO 2 - TRABAJO COLABORATIVO Y GROUPWARE

Lo primero que haremos en este capítulo es definir *aplicaciones colaborativas*, también conocidas como *aplicaciones de groupware*.

Groupware es la tecnología de computación para tratar los problemas del trabajo en grupo. El término groupware es una contracción de las palabras group y software.

Los investigadores de esta área han propuesto varias definiciones; por ejemplo, Malone las define como “la tecnología de información usada para ayudar a las personas a trabajar conjuntamente de manera más efectiva”, implicando, tal vez, que esta nueva área considera a la colaboración más seriamente que los campos de informática tradicionales.

Coleman define el uso de groupware de la siguiente manera: “Groupware permite que equipos de persona y otros paradigmas que requieran de gente puedan trabajar juntos, tanto en tiempo como en espacio”.

La definición de Greenberg es: “Groupware es el software que soporta y aumenta el trabajo en grupo. Es una palabra técnicamente orientada para diferenciar productos “orientados a grupos”, explícitamente diseñados para asistir a grupos de gente a trabajar en conjunto de productos mono-usuarios que ayudan a las personas solo en tareas aisladas desvinculadas con el resto del grupo.”

Lynch, Snyder y Vogel dan su definición haciendo hincapié en las falencias encontradas en los campos tradicionales: “Groupware se distingue del software normal por las suposiciones básicas que hace: concientiza al usuario que es parte de un grupo, mientras que muchos otros softwares intentan ocultar y proteger a los usuarios unos de otros”. Wells aclara que groupware no es simplemente software, es “software y hardware para ambientes interactivos compartidos”.

Ellis, Gibbs y Rein agregan en su definición, el concepto de tarea común: “Groupware son sistemas basados en computadora que apoyan a grupos de personas dedicadas a una tarea u objetivo común y que proveen una interfase a un ambiente compartido.”

En cambio, Peter y Johnson-Lenz, definen groupware sin la necesidad de que las tareas estén asociadas a objetivos específicos. Su primer definición sólo incluía “procesos de

grupo más el software de soporte necesario”, ésta contenía ambos componentes, el humano y el de computación: software en la computadora y software en la gente que lo usa. Ellos ven a los componentes humanos como un “modelo mental” compartido de qué hace el grupo (su propósito) y cómo lo hace (su proceso).

Más adelante, extendieron esta primera definición a “cultura mediada por computadora y procesos de grupo más el software de soporte necesario”. De esta forma, incluyeron más expresamente otros factores culturales, mitos, valores y normas, para que el software de computadora refleje y soporte el propósito, proceso y cultura del grupo. Tal como los procesos en una red cliente servidor, los procesos de computadora y software de grupo unen la máquina y el “hardware” biológico en un único conjunto creativo.

Podemos decir que la cultura mediada por computadora es la personificación en el hyper-espacio de una organización social. Groupware, como cultura es un sistema simple que integra la evolución humana y la de las herramientas.

Con estas definiciones podemos tener una idea sobre las propiedades de las aplicaciones colaborativas, pero no son suficientes para distinguir objetivamente a las aplicaciones colaborativas de las que no lo son. Dewan, Choudhary y Shen dan una definición más precisa y consistente con las definiciones anteriores:

“Una aplicación colaborativa es una aplicación de software que (a) interactúa con múltiples usuarios, esto es, recibe entradas de múltiples usuarios y genera salidas a múltiples usuarios, y (b) acopla las interacciones entre estos usuarios, es decir, permite que la entrada de un usuario influya en la salida mostrada en otro usuario”.

Donde, una entrada es cualquier evento generado por un usuario que es interpretado por la aplicación y una salida es cualquier evento generado por la aplicación que es perceptible por un usuario.

Esta es una definición técnica de una aplicación colaborativa y no una social; dado que los usuarios conectados por la aplicación pueden no sentir realmente que están “colaborando”.

Por ejemplo, según esta definición, una aplicación que permite a los usuarios enviar “mensajes agraviantes” al resto, es una aplicación colaborativa, aunque los usuarios de la aplicación pueden no sentirse que están colaborando.

Esencialmente, esta definición iguala comunicación con colaboración. Pero se puede argumentar que esa comunicación es una condición necesaria pero no suficiente para la colaboración, y así esta definición incluye varias aplicaciones no-colaborativas como la del ejemplo anterior.

Sin embargo, ya que la colaboración está en los “ojos del espectador” o en el “estado mental”, no está claro cómo esta definición se adaptará en un futuro para eliminar las aplicaciones como las de arriba, o, si en realidad, es útil eliminarlas.

2.1 MOTIVACIÓN DE APLICACIONES COLABORATIVAS

Las aplicaciones colaborativas hacen posible el Trabajo Cooperativo Asistido por Computadora o Computer Supported Cooperative Work (CSCW). Pero, ¿cuál es el motivo de proveer soporte de computación para la colaboración, cuando las personas han trabajado juntas durante años sin usar computadoras?

Colaboración Distribuida: las herramientas de computación pueden simular los encuentros cara-a-cara entre usuarios distribuidos, dándoles la ilusión de “estar ahí”. Por ejemplo, una video conferencia y una aplicación de pizarra compartida, conjuntamente pueden simular una reunión cara a cara conducida mediante el uso de una pizarra física. Tal vez, esta sea la mayor motivación de CSCW.

Mas allá de estar ahí: Si todas las aplicaciones colaborativas simularan el “estar ahí”, se soportarían reuniones inferiores a las reales cara-a-cara y está necesidad estaría perjudicando a los usuarios que no pueden estar presentes físicamente. De hecho, estas aplicaciones permiten “más allá de estar ahí” ofreciendo beneficios que no se pueden obtener en las reuniones sin soporte de computación. Permiten disponer de todos los recursos en el ambiente local, estar al mismo tiempo en varias reuniones, formar subgrupos sin estorbar al resto de las personas, hacer comentarios y votaciones en forma anónima, confiar en la computadora el seguimiento de una reunión, salvar el estado de la información de la misma, y continuar con el trabajo privado cuando un ítem de discusión no es de interés.

Hollan y Scott sostienen que las aplicaciones colaborativas no serán verdaderamente exitosas a menos que la gente las elija aún cuando tenga la opción de realizar reuniones cara-a-cara. Para que esto suceda, las personas deben preferir los beneficios de “*mas allá de estar ahí*” sobre los inconvenientes que presentan las reuniones actuales cara-a-cara, que no brindan estos beneficios.

Compartir el estado de la Información y Computación: las aplicaciones colaborativas también son necesarias para permitir a los usuarios visualizar y manipular información que debe ser procesada por la computadora. Por ejemplo, es necesario que un debugger colaborativo permita que múltiples usuarios manipulen conjuntamente un programa.

Sin las aplicaciones colaborativas, los usuarios que desearan manipular conjuntamente información usando la computadora, deberían agruparse alrededor de una estación de trabajo, compartiendo un conjunto físico de dispositivos de entrada y salida. Si bien esta forma de trabajo es actualmente utilizada por algunos grupos pequeños, no es apropiada para colaboraciones en las que intervienen gran cantidad de usuarios.

Se puede argumentar que las aplicaciones que soportan colaboración distribuida y que permiten a los usuarios no estar presentes (*“mas allá de estar ahí”*) son casos especiales de aplicaciones que permiten compartir el estado de la información y computación.

La diferencia principal entre estos dos casos es que en el primero se provee soporte de computación a una tarea colaborativa que, tradicionalmente, se realizaba sin la intervención de la informática. Mientras que en el último caso, se agrega colaboración a una tarea soportada por la computadora, pero que se realizaba individualmente.

A pesar de que éstas pueden ser razones válidas para CSCW, deben ser tomadas con cuidado, ya que esta área todavía está en investigación. Sin embargo, hay varias tendencias que indican que el impacto de CSCW sería grande:

- Algunas herramientas asincrónicas de groupware, como el correo electrónico, la Web y Lotus Notes, han logrado obtener un uso extendido.
- Algunas herramientas sincrónicas, como pizarras compartidas, chats y video conferencias, han sido comercializadas por Intel, IBM, y Microsoft.
- Los estudios de laboratorios sobre algunas aplicaciones colaborativas no comercializadas han sido positivos.
- Las encuestas afirman que gran parte del tiempo (30-70 por ciento) de un trabajador se emplea en asistir a reuniones y que, además, las organizaciones están cada vez más distribuidas.

Se puede sostener que en el futuro todas las herramientas serán colaborativas, ya que muchas de las tareas complejas se informatizarían y se requeriría colaboración. Así, toda aplicación tendría algún estado que más de un usuario puede querer consultar y manipular. ¿Se puede pensar en alguna aplicación que no se quiera compartir en alguna situación?

2.2 CSCW

Las aplicaciones CSCW permiten a los grupos de usuarios, normalmente distribuidos sobre el tiempo o el espacio, resolver cooperativamente una tarea en común por medio de la cooperación de un conjunto de objetos compartidos, tales como documentos, proyectos gráficos, herramientas para construcción de modelos, etc.

El objetivo de los sistemas CSCW es ayudar no sólo a cubrir la acción cooperativa de las tareas sino también la comunicación entre los integrantes del grupo así como también la coordinación de las tareas realizadas individualmente en un grupo. Esto lo hace proveyendo a los usuarios de aplicaciones colaborativas.

Tom Brink lo define de la siguiente manera: "CSCW es el estudio de cómo las personas trabajan conjuntamente haciendo uso de la tecnología de computación. Los temas típicos incluyen el uso de email, hypertext que incluye conocimiento de las actividades de otros usuarios, video conferencia, sistemas de chat, y aplicaciones compartidas de tiempo real, tales como escribir o dibujar colaborativamente."

Otra definición, de acuerdo al Digital Media Laboratory Cybrary es: "CSCW es un campo de investigación multidisciplinario, que incluye la informática, la economía, la sociología, y la psicología. La investigación de CSCW se focaliza en el desarrollo de nuevas teorías y tecnologías para la coordinación de grupos de personas que trabajan juntas."

Por último, Willson da la siguiente definición: "CSCW es un termino genérico que combina, el entendimiento de la forma en que las personas trabajan en grupos, con la ayuda de la tecnología de redes de computadoras y el hardware, software, servicios y tecnologías asociadas."

2.3 DIFERENCIAS ENTRE GROUPWARE Y CSCW

Tendríamos que pensar en groupware como cualquier medio que permite mejorar el trabajo en grupo. Mientras que para aquellas herramientas groupware que consisten en aplicaciones de software, podemos considerarlas como CSCW.

Las siguientes son algunas de las comparaciones realizadas por los distintos investigadores:

"Mientras Groupware se refiere a los sistemas reales basados en computadoras, CSCW estudia las herramientas y técnicas de groupware, así como también sus efectos sociales, psicológicos y organizacionales" (Applied Informatics and Distributed Systems Group at Technische Universitat Munchen)

"Groupware es frecuentemente usado para denotar específicamente la tecnología que las personas utilizan para trabajar conjuntamente, mientras que CSCW se refiere al campo que estudia el uso de esas tecnologías" (Tom Brinck)

Pfeifer, University of Calgary, define CSCW y Groupware apuntando a sus diferencias:

"CSCW es la disciplina científica que motiva y valida el diseño de grupo. Dicho de otra manera, es una ciencia que describe cómo desarrollar aplicaciones de groupware. CSCW, también comprende el estudio y la teoría de cómo la gente trabaja conjuntamente, y cómo el groupware afecta al comportamiento del grupo.

CSCW es una tecnología independiente, pero es socialmente dependiente. Muestra la manera en que la gente interactúa y colabora una con otra, e intenta describir pautas

para desarrollar tecnologías que asistan en el proceso de comunicación.

Groupware es el hardware y software que soporta y aumenta el trabajo en grupo. Las aplicaciones de Groupware no implican reemplazar a las personas en una situación interactiva. Mejor dicho, groupware es una extensión o una herramienta de mejora para los procesos de colaboración. Groupware consiste en productos “orientados a grupos” diseñados para asistir a grupos de personas trabajando juntas. Estos tipos de productos son muy diferentes de las aplicaciones simples de usuario que aíslan al usuario y lo limitan a ejecutar tareas independientes”.

Podemos decir que el objetivo de CSCW es el estudio de cómo las personas trabajan juntas tanto en pequeños grupos como en grandes organizaciones, cómo las aplicaciones de groupware influyen aquellos grupos y organizaciones, y cómo las aplicaciones de groupware pueden aumentar y mejorar la comunicación y coordinación entre los miembros del grupo.

2.4 COMUNICACIÓN, COLABORACIÓN Y COOPERACIÓN

Podemos resumir el propósito de groupware en asistir la comunicación, colaboración y coordinación de las actividades de las personas que trabajan en grupo. Para el éxito de las aplicaciones groupware se necesita proveer soporte para lo que fue denominado “las tres C de workgroup computing”:

Comunicación

Permite el intercambio de por ejemplo ideas, notas durante el transcurso del trabajo. La comunicación mediada o basada en computadora, tal como el correo electrónico, no está completamente integrada con otras formas de comunicación. En primer lugar, el mundo asincrónico, basado en texto, del correo electrónico y de la pizarra de noticias existe separadamente del mundo sincrónico de las conversaciones telefónicas o cara-a-cara.

Las aplicaciones tales como el correo de voz o los programas de conversación empañan un poco esta distinción, aún existen gaps entre ambos mundos. No se puede transferir un documento entre dos números arbitrarios de teléfono, por ejemplo, y no es común originar una comunicación telefónica desde una workstation. La integración de las tecnologías de telecomunicaciones y la computación ayudarán a disminuir dicho gap.

Colaboración

Provee herramientas para trabajar en conjunto en un objeto en común.

Similar a la comunicación, la cooperación es un punto principal en la actividad del

grupo. La colaboración efectiva demanda que la gente comparta información. Desafortunadamente, los sistemas de información actuales, en particular los sistemas de base de datos, van creciendo para aislar a los usuarios unos de otros.

Por ejemplo, consideremos dos diseñadores trabajando con una base de datos CAD. Rara vez, se les permite modificar simultáneamente diferentes partes del mismo objeto y ser concientes de los cambios de los otros; en lugar de esto, ellos deben tomar el objeto y al liberarlo informar los cambios realizados al resto de los diseñadores. Muchas tareas requieren compartir información en un grado aún más fino. Por lo tanto, son necesarios ambientes compartidos que ofrezcan un contexto de grupo actualizado y notificación explícita de las acciones de cada usuario cuando sea apropiado.

Coordinación

La efectividad de la comunicación y colaboración puede ser mejorada si las actividades del grupo son coordinadas. Sin la coordinación, por ejemplo, un equipo de programadores o escritores frecuentemente entrarán en conflicto o repetirán acciones. La coordinación puede ser vista como una actividad en sí misma, como una sobrecarga necesaria cuando varias partes están ejecutando una tarea. Mientras las actuales aplicaciones de base de datos contribuyen a la coordinación de grupos, proveyendo múltiples accesos a objetos compartidos, muchas herramientas de software ofrecen sólo una perspectiva de usuario simple asistiendo muy poco a esta importante función. La coordinación permite el flujo estructurado de los elementos involucrados entre cada tarea, significa armar la agenda y controlar las actividades cooperativas complejas.

2.4.1 COOPERACIÓN Y COLABORACIÓN EN CSCW

De acuerdo a las definiciones anteriores, observamos que los términos CSCW y Groupware significan cosas diferentes para diferentes personas. Por un lado, algunas personas fijan su atención en grupos pequeños que comparten una meta u objetivo común y tienen la necesidad de comunicarse para llevar a cabo esa meta. Para estas personas, CSCW es Trabajo *Cooperativo* Asistido por Computadora.

Por otro lado, algunas personas se concentran en los sistemas organizacionales y en direcciones de proyectos grandes, donde normalmente hay grupos grandes que a menudo tiene objetivos en conflicto. Para estas personas, CSCW es Trabajo *Colaborativo* Asistido por Computadora, o incluso CSW Trabajo Asistido por Computadora.

La primera definición, normalmente proviene del área de Interacción Humano-Computadora o de las Compañías de Software y Telecomunicaciones. Mientras que la última definición, viene del área de Sistemas de Información o de Organizaciones e Instituciones Gubernamentales y Universitarias. (Jonathan Grudin)

CAPITULO 3 - TAILORING

Tailoring es uno de los principales conceptos involucrados en este trabajo, como una breve introducción al mismo, daremos las definiciones encontradas para la palabra *taylor* en el “Oxford English Dictionary”:

- “Tailor” (sustantivo): fabricante de ropa a medida para hombres.
- “Tailor” (verbo): hacer ropa a medida.
- “Tailor” (verbo): construir o adaptar para un propósito específico.

Esta última definición es la que se ajusta a la interpretación de tailoring usada en nuestra tesis.

3.1 SOLUCIONES ADAPTABLES

Una solución no se ajusta a todo: diferentes grupos de personas que interactúan, quienes realizan diferentes tareas en diferentes ambientes con diferentes personas, requieren diferentes tipos de soporte técnico. Por esta razón, el soporte técnico para groupware debe ser ajustable. Los componentes de groupware proporcionan una amplia variedad de tipos de soporte, permitiendo a los usuarios finales ajustar la solución de acuerdo a sus necesidades.

Según Koch y Teege, el crear un nuevo sistema para cada situación específica no es muy eficiente en cuanto al costo. Un sistema compuesto por bloques contruidos por separado, es una solución que permite adaptar adecuadamente el soporte técnico necesario para cada interacción. En segundo lugar, rehusar partes existentes (confiables) puede reducir el esfuerzo requerido para diseñar y desarrollar un sistema, y mejorar la calidad de la solución provista. Por consiguiente, ellos sostienen que los sistemas de groupware deben crearse a partir de unidades funcionales. Al componer un sistema de groupware a partir de los componentes apropiados, según reglas apropiadas, se puede diseñar una solución a medida para cada situación específica.

Aparte del conjunto correcto de herramientas, las personas también necesitan sistemas flexibles que puedan ser modificados por los usuarios finales de acuerdo a sus necesidades, que pueden ser diversas y dinámicas. Las demandas especiales de soporte de trabajo cooperativo hacen que la adaptabilidad sea un aspecto crucial en el diseño de un sistema de groupware.

Las diferencias de complejidad, dinámica, inter/intra individual constituyen la necesidad de diseñar groupware que pueda evolucionar con el tiempo, exhibir diferente comportamiento en diferentes situaciones de uso, y acomodar a las necesidades y preferencias individuales y del grupo.

3.1.1 ADAPTABILIDAD COMO UNA CARACTERÍSTICA CRUCIAL DE GROUPWARE

La adaptación se define como la actividad de modificar una aplicación dentro del contexto de su uso. En la literatura informática, la adaptación es considerada como una característica crucial de los sistemas de groupware debido a que es la manera de obtener una masa crítica y una estabilidad evolutiva. Esta última se refiere a la ambientación de CSCW en la cual están permitidas las desviaciones de los sistemas social y técnico, sin estrés irreparable. Además, los sistemas del groupware proporcionan grandes oportunidades de adaptación, como los descubrimientos experimentales de Mackay muestran que, principalmente, la activación de la adaptación proviene desde orígenes externas, como pueden ser los miembros del grupo o colegas.

Los sistemas adaptables ofrecen a los usuarios finales la posibilidad de ajustar el software a diferentes contextos de uso después que ha sido implementado. Estos contextos de uso deben tomarse en su más amplio sentido: diferentes preferencias personales, diferentes tareas, diferentes grupos de personas y diferentes ambientes. Es obvio que no todas las opciones pueden preverse en la etapa de diseño.

Mørch distingue tres tipos de adaptación, cada una con un impacto diferente en el sistema:

- Adaptación por customización: se refiere a las adaptaciones de la interfase de usuario, significa seleccionar entre un conjunto de opciones de configuración predefinidas.
- Adaptación por integración: significa conectar componentes predefinidos dentro o entre las aplicaciones.
- Adaptación por extensión: significa mejorar las implementaciones incorporando nuevo código de programa.

Los primeros dos acercamientos se refieren a una adaptación cerrada, mientras que el tercero se refiere a una adaptación abierta.

Adaptación abierta

La adaptación abierta o por extensión permite a los usuarios finales hacer cambios profundos al sistema dándoles la misma flexibilidad de los diseñadores originales. Por este motivo, la adaptación abierta puede ser considerada como un diseño continuo. Un ejemplo es la creación de un componente que puede insertarse en un sistema existente, como puede ser el desarrollo de un plug-in para un web browser. Por

supuesto, es cuestionable para sistemas de groupware, si la flexibilidad total es deseable, principalmente porque varios usuarios del sistema de groupware adaptable podrían ser afectados por una operación de adaptación. Un mecanismo de control para un modelo de componente de groupware es diseñar adaptando restricciones, es decir que no son permitidas ciertas etapas de la aplicación o ciertas transformaciones de la aplicación. Además, es cuestionable si todos los usuarios finales deben tener la posibilidad y necesidad de una adaptación abierta. Otro mecanismo de control para un modelo de componente groupware es diseñar reglas de adaptación, los cuales permitan al modelo la distribución de actividades de adaptación dentro de un organismo. Es más, es probable que sólo los usuarios más poderosos, como los administradores de sistema, ingenieros de software e integrantes de mesa de ayuda, ejecuten una adaptación abierta. Los que realicen una adaptación abierta son un grupo de usuarios especializados, con habilidades y conocimientos específicos. Para cuidar la utilidad de un sistema de adaptación abierta, un modelo de componente de groupware debe estar en línea con el modelo mental del grupo de usuarios específicos de la adaptación abierta.

Adaptación cerrada

La adaptación cerrada no permite a los usuarios finales modificar la implementación de la aplicación. La adaptación cerrada puede lograrse cambiando parámetros en las configuraciones predefinidas, modificando la apariencia de los objetos de la interfase de usuario, o conectando diferentes aplicaciones o componentes de alto nivel.

Un mecanismo para un modelo de componente de groupware es diseñar estructuras adaptables, que indiquen explícitamente las partes flexibles de la aplicación. Por supuesto, también las reglas de adaptación son importantes para la adaptación cerrada ya que la adaptación podría afectar a otros usuarios finales.

Se puede prever que la adaptación cerrada requiera menos habilidades y conocimiento del sistema para realizar la adaptación que para realizar la adaptación abierta. El usuario centrado en el diseño de la adaptación cerrada debe poner foco en, qué parámetros deben ser adaptables, qué valores deben ser los predefinidos (= opciones de adaptación) para estos parámetros, y cómo presentar (= apariencia de) estos parámetros y sus valores predefinidos a los usuarios finales.

Perspectiva sobre adaptación centrada en el usuario

Actualmente, en el diseño de sistemas adaptables, se presta mucha atención en las partes a adaptar del sistema: qué aspectos son adaptables, cómo los usuarios finales pueden adaptar dichas partes, etc. A pesar de que es casi ignorada la funcionalidad de lo adaptable y del sistema adaptado; por ejemplo, aspectos tales como cómo los usuarios intercambian entre los modos de usos 'normal' y de adaptación, y cuán utilizables les resultan sus sistemas adaptados.

Sostenemos que ambos aspectos deben ser tomados en cuenta para el desarrollo de sistemas de groupware adaptables. Esto significa que los sistemas de groupware deben ser diseñados de tal manera que provean facilidades de adaptación, mientras que al

mismo tiempo se provea una alta funcionalidad de la adaptación en sí misma y del uso del sistema adaptado.

Adicionalmente, los usuarios finales deben confiar en el sistema, de tal manera de no temer a que se produzcan fallas mientras exploran las opciones adaptables. Es cuestionable si los usuarios finales deben ser conscientes de las diferencias entre la actividad de adaptación y el uso del sistema. Las investigaciones deben identificar si los usuarios finales perciben la adaptación como parte del uso del sistema ‘normal’, o como una actividad separada. Cuando es percibida como una actividad separada, las funciones de adaptación pueden ser presentadas en la interfase de usuario como una barra de herramientas separadas. Por otro lado, cuando es percibida como parte del uso normal, debe ser presentada como parte de la interfase de usuario ‘normal’. Por supuesto, es posible que los usuarios finales consideren algunas funciones de adaptación como parte del uso normal, mientras que consideren otras como parte de la adaptación. Por consiguiente, se debe poner especial atención, durante la etapa de análisis del diseño de la interfase de usuario, a las opciones de adaptabilidad y a la percepción de las mismas por parte del usuario.

Uno de los mayores inconvenientes de la adaptación es la carencia de conciencia de adaptabilidad, lo que significa no conocer qué puede ser adaptado, como tampoco conocer cómo las cosas pueden ser adaptadas.

En los comienzos de los noventa, se encontró que la razón de esto, es la ausencia de documentación, y que los usuarios finales no tienen expectativas en controlar los cambios en el sistema.

Posteriormente, se sugirió que las funciones de adaptación no estaban bastante bien presentadas en la interfase de usuario de las aplicaciones estudiadas. Para ocuparse de la conciencia de adaptabilidad, se introdujo el concepto de ‘*horizonte de adaptabilidad*’ como una nueva característica de gran importancia de los sistemas de groupware adaptables. El horizonte de adaptabilidad es el concepto que provee, al usuario final y potencial adaptador, información sobre las posibilidades de adaptación del sistema y el impacto de la actividad de adaptación sobre la funcionalidad del sistema.

Por supuesto, este concepto es solo aplicable a la adaptación cerrada, dado que las posibilidades para la adaptación abierta son demasiado amplias.

3.1.2 CONTRIBUCIÓN DE LOS COMPONENTES DE GROUPWARE A LA ADAPTABILIDAD

Como vimos anteriormente, la adaptabilidad es de gran importancia para los sistemas de groupware. Las siguientes son las razones de Robert Slagter y Margit Biemans, por las que opinan que los componentes de groupware contribuyen a alcanzar la adaptabilidad:

- Proveen posibilidades para la adaptación abierta: en un sistema de componentes de groupware las dependencias entre las diferentes partes del sistema se hacen explícitamente. Los componentes de software describen explícitamente las interfaces que proveen a sus ambientes y las interfaces que requieren desde sus ambientes. Las dependencias explícitas externas son requeridas para la adaptación abierta, dado que dichas dependencias determinan que tipos de componentes se ajustan al sistema.
- Los componentes son unidades de composición. Por lo tanto, los sistemas basados en componentes permiten la composición de funcionalidad, posibilitando así la creación de sistemas que se ajusten a los requerimientos específicos de un grupo de personas interactuando.
- Los sistemas de componentes de groupware permiten agregar, modificar y eliminar unidades funcionales del sistema. Dependiendo del mecanismo usado, dichas acciones pueden ser ejecutadas durante la inicialización del sistema o en tiempo de ejecución, de compilación o de diseño. Para permitir la composición al usuario final, las acciones mencionadas deben ser posibles durante las etapas de inicialización del sistema o en tiempo de ejecución.

En el contexto ideal, un sistema de groupware debe estar formado por componentes arbitrarios (de tipos adecuados posiblemente creados por diferentes vendedores). Esto es necesario para que la funcionalidad provista se corresponda óptimamente con los requerimientos provocados por la interacción que el sistema debe soportar.

De esta manera, los componentes de groupware pueden proveer a los usuarios finales mecanismos para componer sistemas a partir de unidades funcionales.

Por lo tanto, podemos decir que las tres razones principales de por qué es necesaria la adaptación en sistemas de groupware son:

1. una solución no se ajusta a todo: se necesita diferente soporte técnico dependiendo de la situación específica del grupo interactuante,
2. el Groupware requiere una masa crítica en uso,
3. la adaptabilidad otorga estabilidad evolutiva; es decir, con el tiempo se compensa para distintos contextos de uso: diferentes preferencias personales, tareas, grupos, organizaciones, etc.

3.2 CONTRIBUCIÓN DE LA ADAPTACIÓN AL APRENDIZAJE

En el mundo actual, la tecnología juega un papel importante en los procesos de

aprendizaje. Esto nos obliga a reflexionar sobre los elementos involucrados en su uso y a buscar nuevas formas de enseñar y de aprender eficientemente.

La tecnología informática y de comunicaciones puede cambiar radicalmente la forma de relacionarnos y conseguir información, pero esto no es suficiente. Tenemos que ser capaces no sólo de transmitir información sino también de lograr la asimilación efectiva del conocimiento.

Hay varios trabajos de investigación relacionados con el impacto de la tecnología y los nuevos problemas que pueden surgir de su uso en los procesos de aprendizaje. Pero para entender el impacto y los problemas asociados es importante distinguir entre “el efecto de” la tecnología y “el efecto con” la tecnología. De acuerdo a la perspectiva que se tome para analizar el problema, los resultados pueden cambiar. Se entiende por “efectos de la tecnología” a: qué se ha aprendido y que puede transferirse de aquellas situaciones en las que se trabaja con la computadora y por “efectos con la tecnología” a: qué puede lograr uno en sinergia con una computadora.

Hay que ser escépticos a cualquier predicción en el campo de las tecnologías de la información y cautos a la hora de aplicar la tecnología. Sin embargo, también se considera que estas nuevas herramientas facilitarán la adquisición de las habilidades de aprendizaje, la elaboración de estrategias propias de aprendizaje, el reconocimiento de los estilos típicos de aprendizaje, la selección del recurso de aprendizaje apropiado, así como el refuerzo del valor de las habilidades de reflexión. Se puede prever que los recursos digitales serán más interactivos, adaptables y amistosos, promoviendo la creatividad del usuario y la integración de la experiencia.

En nuestra tesis está presente el uso de la tecnología para integrar el aprendizaje individual y el aprendizaje colaborativo a través de un modelo adaptativo.

Actualmente se han desarrollado muchos ambientes de aprendizaje basados en tecnologías de la información y de comunicaciones. Unos hacen más énfasis en el aprendizaje individual, otros en el aprendizaje colaborativo y existen algunos que integran ambos aspectos. La meta del apoyo a la colaboración adaptativa es conformar grupos de colaboración.

Oppermann plantea que el concepto de la adaptación ha sido una de las cuestiones más importantes de la investigación dentro de los sistemas de aprendizaje en los últimos años. La investigación ha mostrado que el uso de adaptación puede proveer mejores entornos de aprendizaje en tales sistemas. Se han desarrollado dos clases de sistemas para dar soporte al usuario en sus tareas:

- *Sistemas Adaptables*: permiten al usuario cambiar ciertos parámetros del sistema y adaptar de esta manera su comportamiento.
- *Sistemas Adaptativos*: se adaptan al usuario automáticamente basándose en las suposiciones que el sistema realiza de las necesidades del usuario.

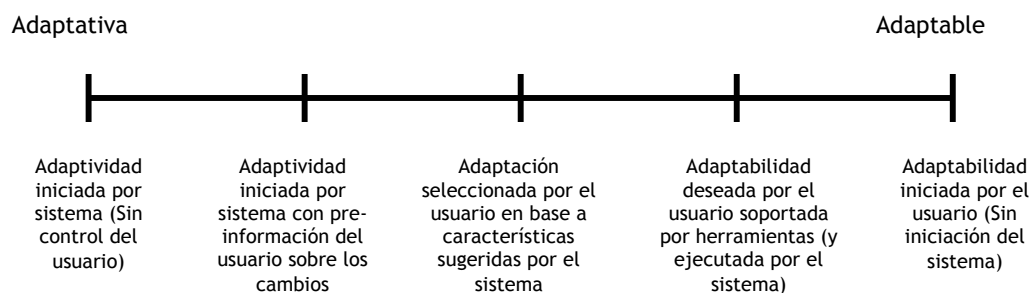


Figura 3.1 - Espectro de adaptación en sistemas de computadores

3.2.1 APRENDIZAJE INDIVIDUAL Y APRENDIZAJE COLABORATIVO

El Aprendizaje Individual está orientado a satisfacer necesidades del usuario, que pueden variar en el tiempo, la forma, el contenido y el volumen. Esto determina la necesidad de que los ambientes desarrollados para apoyar el Aprendizaje Individual sean flexibles, amigables y tengan incorporado los conceptos de adaptación. La valoración que un usuario particular tendrá de un sistema está determinada por la habilidad del sistema para facilitarle su aprendizaje.

En las teorías de aprendizaje queda claro que la interacción es un factor catalizador del proceso de aprendizaje. Pero la creación de conocimiento y la asimilación del mismo es siempre un proceso individual.

El Aprendizaje Colaborativo está orientado a la generación de conocimiento y lo podemos definir como: “Co construcción de conocimiento, y mutuo compromiso de los participantes”.

En términos genéricos, se define colaboración como: cualquier actividad que un par o más de individuos efectúan juntos.

El Aprendizaje Colaborativo intenta eliminar los problemas de aislamiento y de soledad que tienen los usuarios al interactuar con ambientes de aprendizaje individual sin la presencia de grupos de trabajo. La colaboración es un catalizador de conocimientos y muchos de los avances están orientados a la socialización del proceso de aprendizaje.

Varias de las metodologías y herramientas que usan la tecnología (hardware y software) para apoyar el Aprendizaje Colaborativo han surgido bajo el nombre de “Aprendizaje Colaborativo Soportado por Computador” (CSCL-Computer Supported Collaborative Learning). Pero en este campo no sólo se ha investigado la tecnología sino también los aspectos sociales, psicológicos, organizacionales y los efectos en el aprendizaje. Por los resultados de las investigaciones realizadas se puede deducir que el solo hecho de utilizar tecnología no soluciona los problemas asociados al proceso de aprendizaje.

El proceso de aprendizaje, donde intervienen grupos de usuarios, tiene dos secuencias: Secuencia de Transferencia de Conocimiento y Secuencia de Colaboración. En la

primera es donde los usuarios intercambian conocimientos y en la segunda es donde se sintetiza y aplica el conocimiento recibido.

Nuestro trabajo es un sistema que integra el aprendizaje individual con el aprendizaje colaborativo, y en el que ambos aprendizajes son adaptativos. Una característica importante es que los modelos utilizados para la adaptación en el sistema de aprendizaje individual, son los mismos que se utilizarán para conseguir la adaptación durante el aprendizaje colaborativo.

Debido a que el trabajo colaborativo involucra a un conjunto de dos o más personas, el groupware no solo debe ser aceptado por una única persona, sino que debe ser aceptado por todos los miembros del grupo o al menos por un número considerable de ellos.

El principal acercamiento al problema de la aceptación es la adaptación del sistema y de los procesos soportados por el sistema. La adaptación también enfoca a los diferentes requerimientos de soporte, los cuales resultan del uso del sistema en distintos proyectos y fases de los mismos, tal como fases de actividades sincrónicas y asincrónicas.

Para usar la adaptabilidad de las formas mencionadas, es de gran importancia ofrecer adaptación a los usuarios que ejecutan procesos de grupo soportados por el software.

La adaptación en sistemas CSCW afecta de diferentes maneras a los integrantes de un grupo. Esto puede llevar a una tensión entre un individuo y el grupo, o entre diferentes individuos. La adaptación realizada por algunos puede aumentar la sobrecarga de trabajo de otros. Como en todo grupo de trabajo es necesario identificar a los “contenedores de apuestas” (stakeholders) del proceso de adaptación. Una posible solución para moderar los diferentes intereses pueden ser los derechos de adaptación. Además de identificar los diferentes “contenedores de apuestas”, existe la necesidad de adaptar el software al ambiente organizacional tal como a las necesidades individuales.

En los últimos años se fue demostrando que la flexibilidad y la adaptatividad son requerimientos claves y críticos para el éxito de los sistemas de software en general y para los sistemas CSCW en particular. Para estos últimos, estas propiedades son aún más importantes, ya que se debe satisfacer a muchas personas al mismo tiempo.

En los sistemas CSCW la adaptatividad se refiere a los requerimientos de las tareas del grupo y de la organización en la cual el sistema es usado, así como también, a las preferencias personales, y a las necesidades de los usuarios individuales y de los grupos de usuarios.

Si los requerimientos de flexibilidad y adaptatividad son ignorados, el uso del producto CSCW nunca alcanzará la masa crítica de usuarios necesaria para que el software sea útil y provechoso. Por tal motivo, es crucial para su éxito, la aceptación del software por todos los miembros del grupo, en lugar de solo por algunos individuos como sucede

en un software monousuario donde un único usuario puede beneficiarse con el uso del mismo.

De todos modos, no siempre es fácil alcanzar dicha aceptación, dado que los miembros del grupo usualmente tienen diferentes preferencias (como la elección de un editor de texto específico), diferentes experiencias, y frecuentemente tienen habilidades y educaciones heterogéneas.

Además de los diversos requerimientos de los miembros de un grupo, el uso de sistemas CSCW en distintos proyectos resulta en diferentes requerimientos concernientes a su funcionalidad. Más aún, dentro de un mismo proyecto, se distinguen diferentes etapas de uso, posiblemente, con diferentes requerimientos; por ejemplo, períodos de actividad sincrónica, cuando el grupo coopera conjuntamente, y períodos de actividad asincrónica, cuando los miembros del grupo no tienen contacto o trabajan en diferentes horarios.

A la hora de realizar una adaptación se deben tener en cuenta las características y preferencias del usuario. Bajo este concepto se incluye todo lo referente a *preferencias y conocimientos* del usuario. Por *preferencias* entendemos todo lo relativo a la manera de recibir la información, a las características de aprendizaje, a las características personales (si es un usuario extrovertido o introvertido), etc. Y por *conocimiento*, todo lo relativo a conocimiento previo o evolutivo.

CAPITULO 4 - EVOLUCIÓN COLABORATIVA DE CONFIGURACIONES DE TRABAJO

4.1 AMBIENTES DE TRABAJO

Como mencionamos en el capítulo 1, podemos observar como los seres humanos tienden a adaptar los ambientes que frecuentan de acuerdo a sus necesidades. Este proceso de organización de los objetos que nos rodean es casi inconsciente y favorece a agilizar el trabajo diario.

Sólo con dar una mirada a nuestro alrededor, podemos ver que existen distintos y diversos ambientes de trabajo. Por ejemplo, en una visita al odontólogo, observamos que éste tiene organizadas sus herramientas de tal manera que aquellas que utiliza frecuentemente las tiene más cerca, y las que son de uso más esporádico están ubicadas en lugares más apartados. Por lo general, el torno está al alcance de su mano, mientras que los utensilios para realizar un tratamiento de conducto no siempre están tan a mano.

Además, cuenta con información sobre sus pacientes, la cual puede estar en fichas de papel o guardada de manera digital en una computadora.

También podemos notar que el ambiente de trabajo de un odontólogo novato difiere del de uno con más experiencia. El experto puede tener más variedad y cantidad de herramientas, producto de su trayectoria en la profesión, mientras que el novato puede comenzar sólo con las herramientas básicas.

Por otro lado, podemos observar que el ambiente de trabajo de un contador es muy distinto al del odontólogo. En general, trabaja sobre una computadora dónde tiene todos los aplicativos necesarios para realizar su trabajo cotidiano. También puede disponer de una biblioteca con libros y leyes concernientes a su profesión. Y análogamente al odontólogo, dispone de carpetas con documentación de sus clientes.

A partir de estos dos ejemplos, notamos que un ambiente de trabajo esta dado por la disposición de las herramientas e información necesarias para desarrollar una actividad determinada, y que los seres humanos, independientemente de la actividad que

desarrollen, tienden a organizar sus ambientes de trabajo de acuerdo a sus necesidades, experiencia y estilo de trabajo.

Este proceso de organización de los objetos que nos rodean es casi inconsciente y favorece a agilizar el trabajo diario.

EJEMPLO: TALLER MECÁNICO

Si tomamos como ejemplo el ambiente de trabajo de un mecánico, vemos que éste organiza su ambiente de trabajo ubicando aquellas herramientas e información (manuales, libretas de teléfonos, libro de cuenta corriente, etc.) que usa más a menudo en lugares fáciles de acceder y guardando las menos usadas en lugares más apartados.

Si observamos con más detalle, los ambientes de trabajo de dos mecánicos con igual especialización, podemos ver que ambos tienden a utilizar las mismas herramientas o, por lo menos coinciden en el uso de muchas de ellas, y las organizan y distribuyen de manera similar. Con lo cual, ambos mecánicos podrían compartir el mismo lugar de trabajo sin mayores inconvenientes.

Pero si ahora observamos el ambiente de trabajo de un mecánico que se especializa en *carburación* podemos ver que difiere del ambiente de trabajo de otro mecánico que se especializa en *tren delantero*. Si bien pueden coincidir en el uso de muchas de las herramientas, la disposición de éstas no es la misma, dado que la frecuencia de uso es diferente en cada especialización. Por ejemplo, el especialista en carburación usa habitualmente el saca bujías, mientras que el otro especialista la puede usar, pero no tan a menudo.

Con lo cual, si estos dos mecánicos, trabajaran en el mismo taller en distintos horarios dispondrían de las herramientas de trabajo, pero no habría forma de adaptar el lugar a las necesidades de cada uno. Lo que implicaría que antes de comenzar a trabajar, cada mecánico debería perder tiempo en localizar y organizar las herramientas necesarias para sus actividades.

PARALELISMO CON UN AMBIENTE DE TRABAJO INFORMÁTICO

Podemos hacer una similitud entre el taller del mecánico y una computadora, donde el ambiente de trabajo sería la PC y las herramientas serían las aplicaciones y/o documentos. De igual forma, una persona dedicada a la informática o que utiliza una PC como herramienta de trabajo principal, organiza las aplicaciones y/o documentos más usados en lugares de fácil y rápido acceso.

En el ámbito de la informática, también existen diversas especializaciones, por ejemplo, programadores, analistas, diseñadores, etc. Las mismas pueden coincidir en el

uso de muchos de los recursos, pero utilizarlos con distinta frecuencia. Por ejemplo, un analista puede trabajar habitualmente con el Project para la planificación de las tareas del grupo y la registración del avance de las mismas, mientras que el programador y el diseñador, pueden tener la necesidad de consultar dicha planificación, pero no trabajan cotidianamente sobre la misma.

Dentro de cada especialización hay personas más experimentadas que otras. Si tomamos el caso de los programadores, podemos ver como los más novatos están al tanto de los últimos avances de la tecnología, de las últimas versiones de software que salen al mercado y de toda nueva aplicación innovadora. Mientras que un programador con más experiencia, tiende a estar especializado en áreas, lenguajes o herramientas más tradicionales en las que desarrolla su trabajo.

EVOLUCIÓN, ADAPTACIÓN Y COLABORACIÓN

A menudo los usuarios que realizan actividades similares o tienen la misma especialización, utilizan los mismos documentos y/o aplicaciones para realizar sus tareas habituales. A medida que avanzan en sus tareas, van organizando y distribuyendo dichos documentos y aplicaciones en su entorno de trabajo, de acuerdo a la frecuencia de uso de los mismos, ubicando los más usados en lugares de rápido acceso, como puede ser el escritorio de Windows.

Con lo cual, dentro de un grupo de trabajo, cada persona va consumiendo de forma aislada parte de su tiempo, esfuerzo y energía para obtener un ambiente de trabajo adaptado a sus necesidades. Sin tener conocimiento que sus necesidades son similares a las de otros miembros del grupo que realizan las mismas tareas, lo que provoca un notable aumento del tiempo y del esfuerzo aportado por cada uno de los integrantes.

Dicho tiempo, esfuerzo y energía podría reducirse, si existiera comunicación y colaboración entre los miembros del grupo. Dado que tanto los usuarios novatos como los experimentados, podrían aprender uno de otros, compartiendo sus conocimientos e intercambiando experiencias personales.

Por esta razón, en los ambientes de trabajo cotidiano, como pueden ser los ambientes científicos o creativos, está creciendo aceleradamente la necesidad de la colaboración en equipo y de ambientes de trabajo flexibles.

El trabajo colaborativo que se realiza en ambientes creativos, de diseño o corporativos, a menudo implican trabajar en equipos cambiantes de diferentes tamaños y trabajar con diferentes herramientas o diferentes combinaciones de las mismas. Además, con el transcurso del tiempo, el producto del trabajo de un equipo evoluciona y aumenta su complejidad, por lo cual se requieren nuevas y diferentes herramientas (por ejemplo, nuevas herramientas de estructuración, editores para los nuevos tipos de contenidos, etc.). Otro factor de influencia para esta necesidad de cooperar es el advenimiento de

las "organizaciones virtuales" o "empresas extendidas": estructuras impulsadas por tareas que abarcan varias unidades organizacionales dentro de una compañía o, incluso entre más de una compañía, están potencialmente distribuidas en una escala global, formadas con el objetivo concreto de generar un cierto producto o de realizar un proyecto específico.

Como vimos en el capítulo 2, las aplicaciones CSCW permiten a tales grupos de usuarios (normalmente distribuidos en tiempo y/o espacio) resolver cooperativamente una tarea común cooperando sobre un conjunto de objetos compartidos, como pueden ser un documento, un dibujo gráfico, un modelo de construcción, etc. Sobre estos objetos toma lugar la cooperación.

Por otro lado, se puede ver la tendencia de que los oficinistas se están volviendo cada vez más y más movedizos. Este incremento de la movilidad de los trabajadores y la flexibilidad de la estructuración del trabajo condujeron al desafío de permitir, dentro de procesos colaborativos, usuarios con diferentes roles, usando un amplio rango de diferentes plataformas con diferentes capacidades. Los dispositivos informáticos móviles tienen varias limitaciones, como por ejemplo memoria, display, etc. Así, los sistemas que soportan colaboración necesitan ser extensibles y adaptables a situaciones y ambientes de trabajo variantes, como también necesitan permitir la reusabilidad y la combinación de las herramientas para diferentes ambientes de equipo. Los miembros del equipo necesitan poder extender y adaptar su ambiente de trabajo para adecuarlo a sus necesidades cambiantes.

El desarrollo basado en componentes ha evolucionado de manera prometedora para permitir la reusabilidad, extensibilidad y adaptación de los ambientes de aplicación.

4.2 INTRODUCCIÓN AL IWA

Actualmente, la mayoría de los trabajos necesitan de la ayuda de una computadora para ser llevados a cabo de manera más ágil, rápida y eficiente. De esta forma, como mencionamos en la sección anterior, las personas que trabajan diariamente con una computadora tienden a configurar su entorno de trabajo con las herramientas y documentos que usan más frecuentemente. Esto se percibe, por ejemplo, al observar como los usuarios de entornos X-Windows adaptan sus escritorios y menús de acceso rápido para llevar a cabo las tareas que más comúnmente realizan.

Hoy en día, el esquema más avanzado para modelar el entorno de trabajo está basado en identidades. A medida que un usuario va adaptando su entorno de trabajo va obteniendo una configuración de acuerdo a sus necesidades. De esta forma, una configuración está generalmente determinada por una configuración básica a la que se le agrega la configuración dependiente de la identidad del usuario. Esta va

evolucionando a medida que el usuario ataca nuevas tareas o desempeña distintos roles. Así, cada usuario modifica su entorno de trabajo en forma aislada e independiente.

Por otro lado, también observamos que una porción significativa de las actividades de una persona ocurren dentro de un grupo, en lugar de en un contexto individual. Por este motivo, como vimos en el capítulo 2, cada vez son más los equipos de trabajo que utilizan aplicaciones groupware para llevar a cabo sus tareas, aprovechando todos los beneficios y ventajas que estas aplicaciones les brindan.

Los grupos de trabajo pueden tener como objetivo comunicar, cooperar, coordinar, resolver problemas, competir o negociar, en los cuales más de un usuario puede tomar el mismo rol.

En este contexto resulta interesante la posibilidad de configurar el entorno de trabajo en base a los distintos roles o actividades que el usuario desempeña. De esta forma, cada usuario podría compartir su configuración con el resto de los usuarios del grupo que estén desarrollando su misma actividad y al mismo tiempo aprender de ellos, optimizando así su entorno de trabajo. Cada una de las personas tendría la configuración del entorno de trabajo asociada a su identidad, y a la del o los roles que esté desempeñando. Para lograr esto, necesitamos una herramienta que nos permita:

Mantener el ambiente de trabajo de cada individuo

- Que se cargue automáticamente; es decir, cuando el usuario ingresa a su entorno de trabajo, lo encuentra de la misma forma que lo dejó la última vez.
- Que lo pueda adaptar, agregando y/o eliminando las herramientas de acuerdo a los requerimientos de su trabajo.

Permitir la adaptación colaborativa

- Compartir cambios intercambiando conocimientos y experiencias personales; permitiendo que los cambios afecten a todos los usuarios de la misma especialidad.
- Facilitar la comunicación entre los miembros del grupo para discutir y negociar los cambios.

4.2.1 IDEAL WORKSPACE AGENT

Como vimos anteriormente, en un grupo de trabajo asistido por computadora, los usuarios que desempeñan un mismo rol, podrían compartir su configuración con el resto de los usuarios del grupo que estén desarrollando su misma actividad. Cada una de las personas tendría la configuración del entorno de trabajo asociada a su identidad, y a la del o los roles que esté desempeñando.

A medida que los usuarios avanzan en sus labores y atacan nuevas tareas, van

necesitando del uso de nuevas y distintas herramientas. Con lo cual, los usuarios que desempeñen un mismo rol deberán lograr que la configuración evolucione de manera conjunta y colaborativa.

Nuestro trabajo tiene como objetivo principal obtener una herramienta de uso sencillo y práctico, que se adapte a las necesidades de los grupos de trabajo asistidos por computadora, en el que las personas que realizan las mismas tareas son identificadas bajo un mismo rol, permitiéndoles evolucionar colaborativamente a una Configuración de Trabajo “ideal”. A esta herramienta la denominamos Ideal Workspace Agent (IWA).

4.3 CONCEPTOS INVOLUCRADOS

Para desarrollar un ejemplo concreto de lo planteado, elegimos extender la funcionalidad que provee el Sistema Operativo Windows para el manejo de entornos de trabajo, los cuales están compuestos por un escritorio, menú de inicio, favoritos, etc.

Dado que esta plataforma nos permite adaptar el ambiente de trabajo para un usuario y mantenerlo consistente en los sucesivos ingresos al mismo, nuestra herramienta deberá extender esta funcionalidad para cubrir los aspectos de colaboración y comunicación necesarios para poder obtener un entorno de trabajo compartido por los miembros de un grupo que desarrollen la misma actividad, y que pueda evolucionar colaborativamente.

En el capítulo siguiente veremos en detalle como implementamos el caso elegido, pero antes es necesario ver los conceptos principales en los que nos basamos.

Configuración de Trabajo

En el IWA la configuración de trabajo de cada usuario comprende los distintos elementos en los cuales el usuario puede agregar, eliminar y organizar las herramientas y documentos que utiliza con mayor frecuencia. Cada uno de estos elementos está formado por una colección de herramientas y documentos que va variando a medida que el usuario avanza en sus tareas.

Por ejemplo, en los entornos X-Windows entre los elementos que componen la configuración de trabajo podemos encontrar el escritorio, el menú de inicio, el menú de favoritos, barra de acceso rápido, etc.

Roles

Como mencionamos en el capítulo 2, un rol es un conjunto de privilegios y responsabilidades atribuidas a una persona. Dentro de un grupo de trabajo, los individuos pueden tomar diferentes roles de acuerdo a la actividad que desempeñan.

En el IWA, cada rol tiene asignado una configuración de trabajo que está compuesta por los recursos necesarios para desarrollar las tareas correspondientes a dicho rol.

Además, permite la manipulación de roles, ya sea crear nuevos cuando surjan nuevas tareas, o eliminar existentes debido a que ya no son necesarios para el trabajo que está llevando a cabo el grupo.

Los usuarios también pueden cambiar de un rol a otro, o tener asignado más de uno a la vez si así lo requieren las actividades que desempeñan.

Entorno de Trabajo

En el IWA, existen configuraciones de trabajo asociadas a la identidad de cada usuario y asociadas a cada uno de los roles. De esta manera, el entorno de trabajo de un usuario está formado por un conjunto de configuraciones de trabajo, este conjunto está compuesto por la configuración asociada a la identidad del usuario y por las configuraciones asociadas a la del o los roles que éste desempeña.

Evolución colaborativa del entorno de trabajo

A medida que el usuario avanza en sus actividades o ataca nuevas tareas, puede percibir que con frecuencia utiliza algún recurso que no se encuentra disponible en su entorno de trabajo. Ante esta necesidad, podrá incorporar la nueva herramienta a su entorno de trabajo para agilizar sus labores cotidianas. Para esto, el usuario evaluará si el nuevo recurso también es de utilidad para el resto de los usuarios que cumplen el mismo rol, o si solo es útil para su trabajo específico. En este último caso, el usuario agregará el recurso a su configuración de trabajo personal, obteniendo de esta manera una configuración ideal asociada a su identidad.

En cambio, si el usuario decide que el nuevo recurso ayudará en la realización de las tareas llevadas a cabo por alguno de los roles que desempeña, intentará incorporarlo a la configuración del rol correspondiente. Para esto, deberá iniciar una comunicación con el resto de los usuarios explicando cómo el uso de esta nueva herramienta los asistirá en la ejecución de sus tareas cotidianas. De esta manera, se da comienzo a un debate entre los integrantes del grupo, donde cada uno de estos expresa su opinión acerca de la incorporación propuesta. Dependiendo del consenso obtenido de los miembros del rol a lo largo del debate, se determinará la incorporación o no de la herramienta a la configuración de trabajo asociada a dicho rol. Así, las configuraciones de trabajo asociadas a los roles evolucionan colaborativamente a una configuración “ideal” para dicho rol.

Por otro lado, cuando un usuario ve la necesidad de incorporar un nuevo rol al sistema, de manera similar a la anterior, iniciará una nueva comunicación con el resto de los usuarios para negociar sobre la incorporación del nuevo rol.

La eliminación de un recurso o de un rol existente, también se acordará mediante un proceso de negociación/discusión como el descrito anteriormente.

En consecuencia, el entorno de trabajo de los integrantes del grupo evoluciona conjunta y colaborativamente hacia una configuración “ideal”.

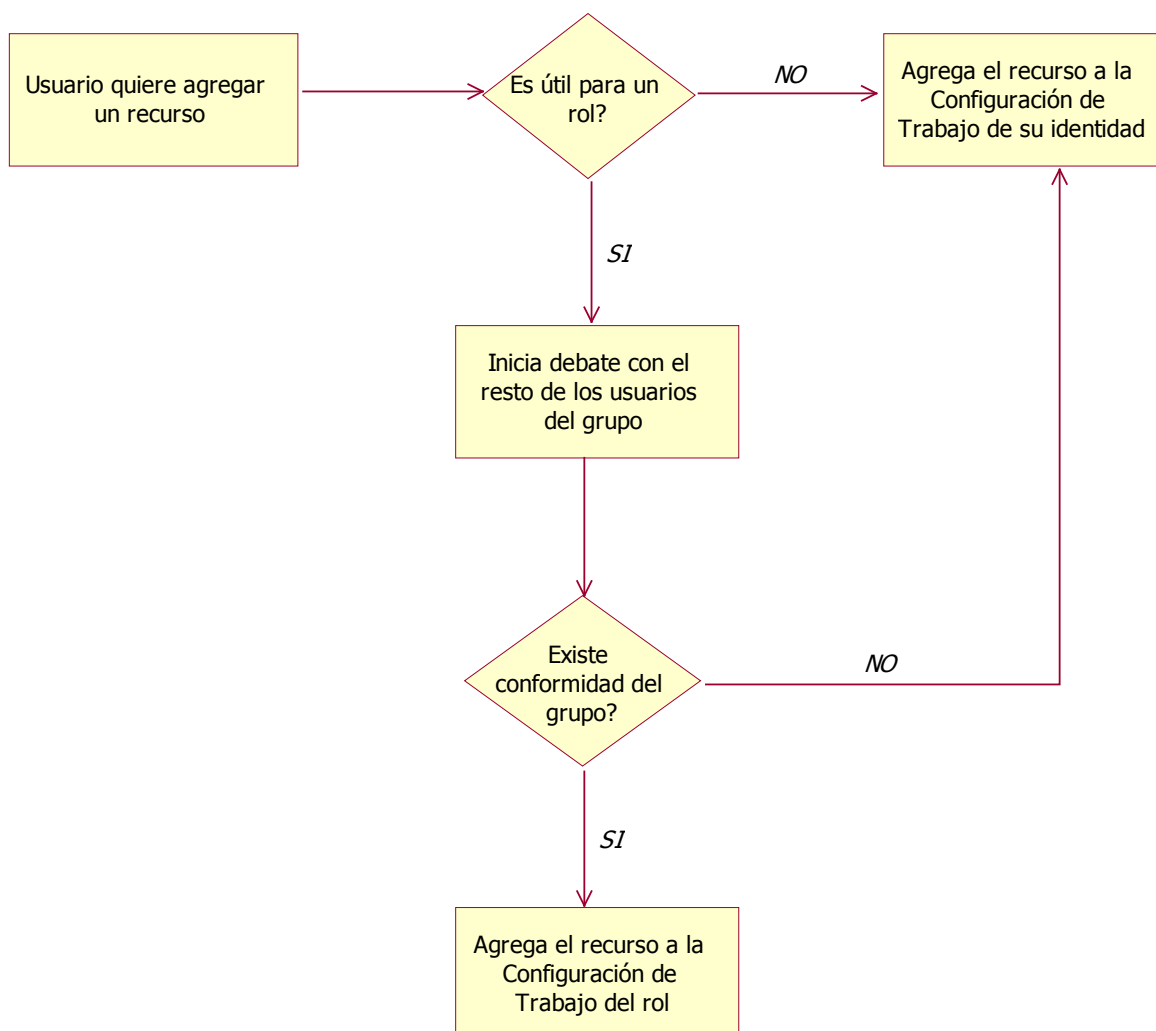


Diagrama 4.1 Evolución colaborativa del entorno de trabajo

4.4 LOS ROLES Y EL ENTORNO DE TRABAJO

La primera vez que un usuario ingresa al IWA, se le generará la configuración de trabajo asociada a su identidad. Esta configuración inicial se creará a partir de los elementos que componen el entorno de trabajo del usuario en ese momento. De esta forma, el entorno de trabajo para ese usuario en el IWA se corresponde con su entorno

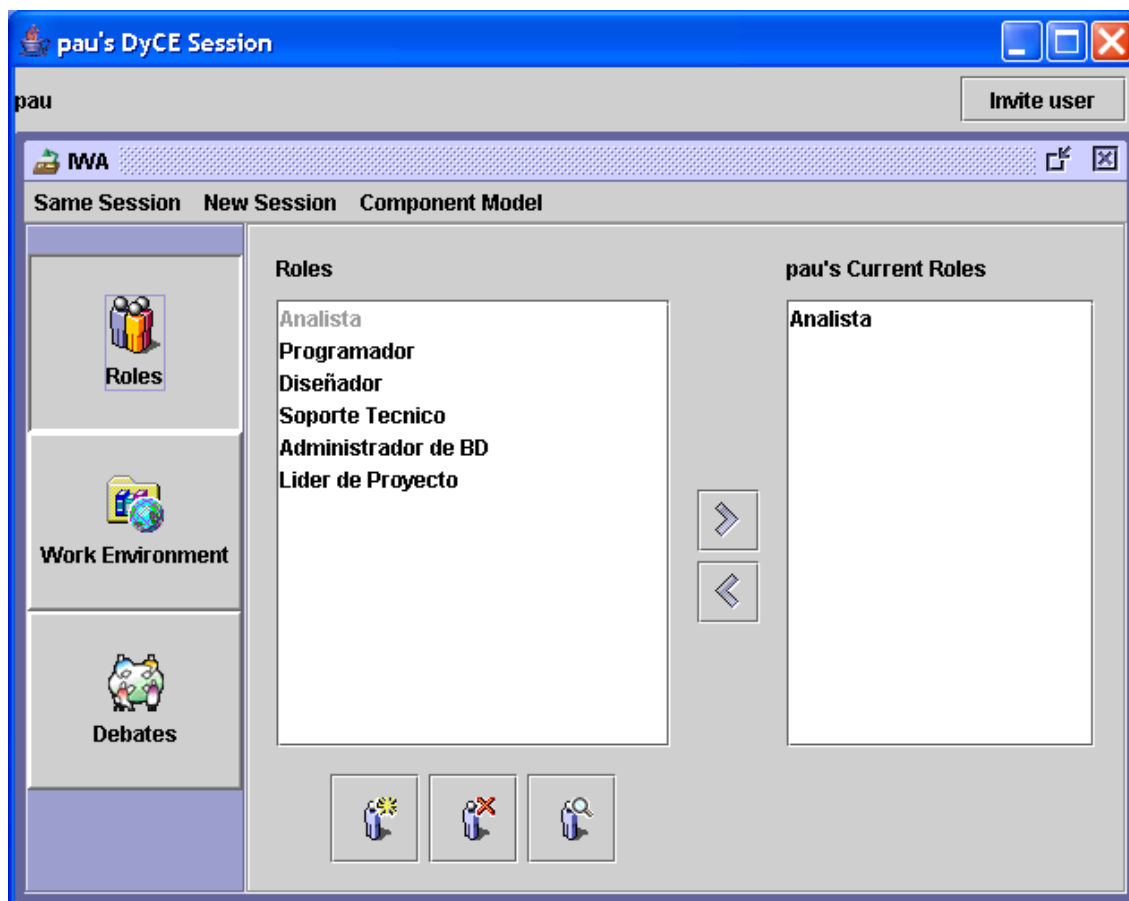
de trabajo en el X-Windows.

En las siguientes oportunidades, se le actualizará el entorno de trabajo de X-Windows con el entorno de trabajo guardado en el IWA para ese usuario. Este entorno de trabajo estará compuesto por la configuración de trabajo asociado a su identidad más la suma de las configuraciones asociadas a los roles con los que el usuario trabajó la última vez.

Todo usuario del IWA, podrá elegir manualmente con qué rol o roles desea trabajar, cambiar de un rol a otro, o tener asignado más de uno a la vez si así lo requieren las actividades que desempeñan.

Cuando un usuario selecciona un rol para trabajar, éste se agregará a su lista de roles y se actualizará su entorno de trabajo con los documentos y herramientas correspondientes a la configuración de trabajo del rol elegido.

Cuando un usuario no va a realizar más tareas relacionadas a un rol, podrá “eliminarlo” de su lista de roles. Esto provocará la actualización de su entorno de trabajo, borrándose del mismo la configuración correspondiente a dicho rol.



Pantalla 4.1 Roles

4.4.1 MANEJO DE ROLES

Cuando surjan nuevas tareas, cualquier usuario podrá sugerir la creación de un nuevo rol, o la eliminación de alguno existente debido a que ya no es necesario para el trabajo que está llevando a cabo el grupo. También podrá sugerir la incorporación o eliminación de un recurso a la configuración de trabajo asociada a un rol.

El usuario interesado en eliminar un rol, deberá debatir on-line con el resto del grupo sobre la eliminación de dicho rol. Sólo se permitirá eliminar un rol que no esté siendo utilizado por ningún otro usuario; dicha eliminación se realiza de manera lógica, para disponer de información sobre la historia de los roles por los que pasó cada usuario.

En el caso del alta de un rol, el usuario también podrá debatir on-line con el grupo para determinar la creación o no del nuevo rol.

En ambos casos, la discusión/negociación llevada a cabo por el grupo se guardará como parte del rol, para poder ser consultada o reanudada en cualquier momento.

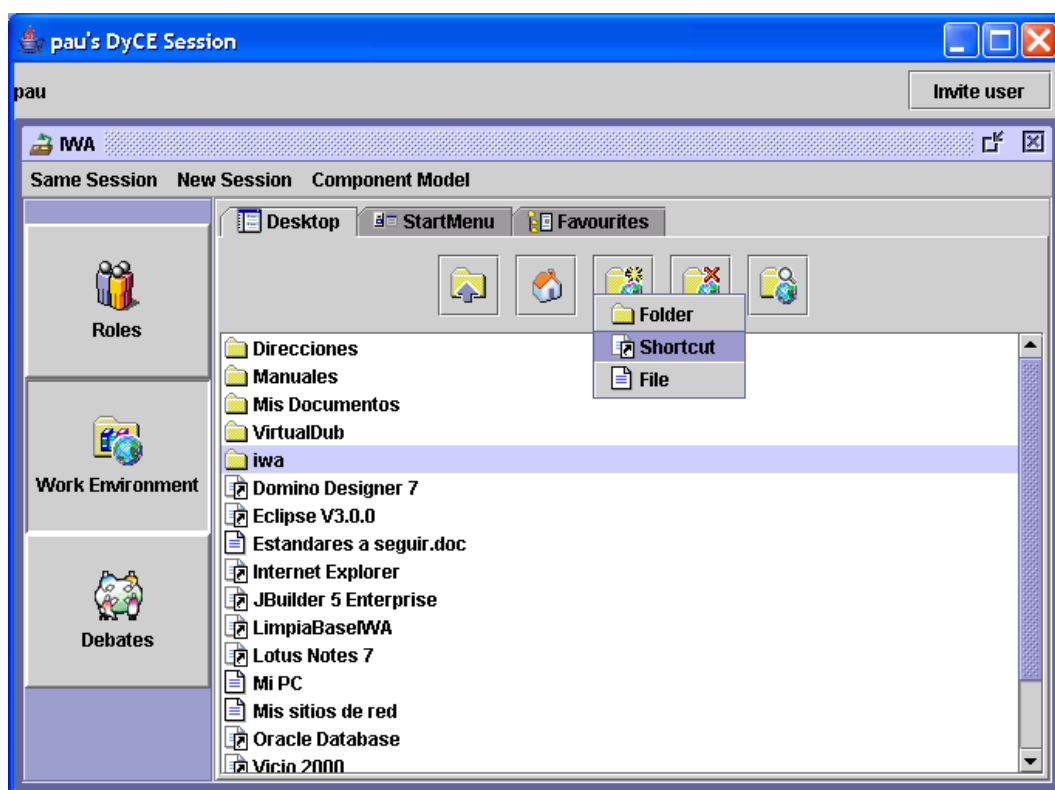
La modificación de la configuración de trabajo asociada a un rol, es decir la incorporación o eliminación de un recurso a la misma, es la base de la evolución colaborativa de una configuración de trabajo que se detalla en la siguiente sección.

4.5 EVOLUCIÓN COLABORATIVA DE UNA CONFIGURACIÓN DE TRABAJO

INCORPORACIÓN DE UN RECURSO A UNA CONFIGURACIÓN DE TRABAJO

Cuando un usuario ve la necesidad de incorporar un nuevo recurso a su entorno de trabajo, y no está desempeñando ningún rol, el IWA lo agregará directamente a la configuración asociada a su identidad.

En caso que el usuario tenga asignado uno o más roles, le presentará una ventana desde la cual deberá seleccionar la configuración en la que desea agregar el nuevo recurso. Si cree que éste ayudará en la realización de las tareas llevadas a cabo por alguno de los roles que está desempeñando, elegirá la configuración asociada a dicho rol. En cambio, si cree que sólo lo ayudará en sus tareas personales, elegirá la configuración asociada a su identidad.



Pantalla 4.2 Incorporación de un recurso

ELIMINACIÓN DE UN RECURSO DE UNA CONFIGURACIÓN DE TRABAJO

Cuando un usuario desea eliminar un recurso de su entorno de trabajo, y no está desempeñando ningún rol, el IWA lo eliminará directamente de la configuración asociada a su identidad.

En caso que el usuario tenga asignado uno o más roles, le presentará una ventana con una lista con los roles que está desempeñando que contengan a dicho recurso como parte su configuración, y con su identidad, en caso que también lo contenga. La selección que haga el usuario dependerá de a qué configuración desea que afecte el cambio.

Remove Resource

D:\Documents and Settings\USUARIO\Escritorio\Programador

Name: Visio 2000.lnk

Target: ts and Settings\USUARIO\Mis documentos\marti.zsh

The new resource will be added to:

☐ User Paula

☒ Role Programador

Debates

Operation	Init Date	End Date	Initiator	
Create	08/02/2005	08/02/2005	Paula	...

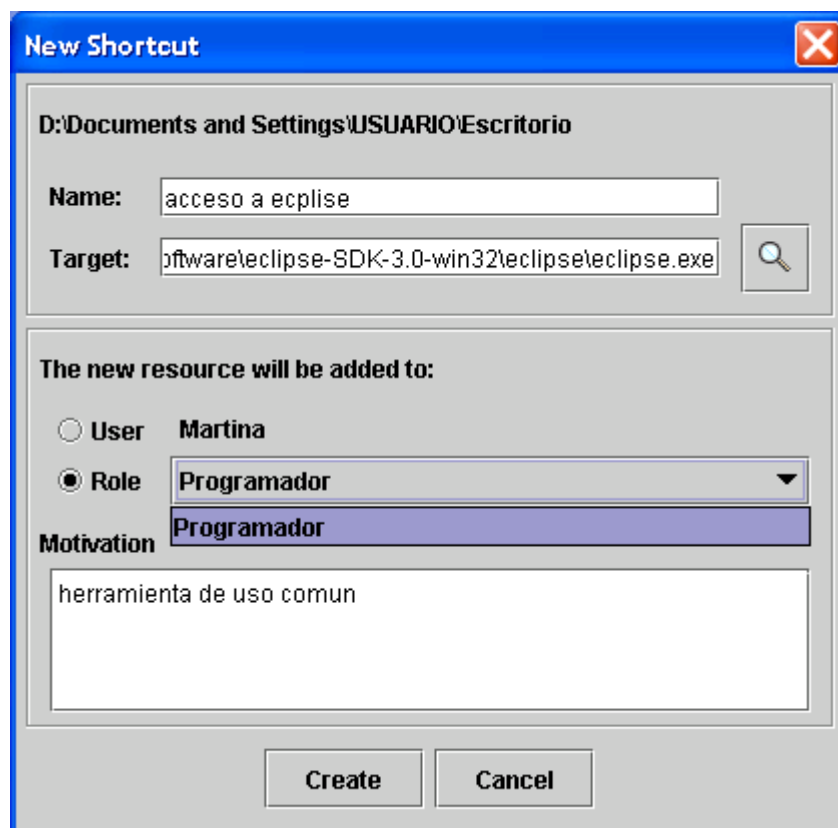
Remove **Cancel**

Pantalla 4.3 Eliminación de un recurso

Tanto en la incorporación como en la eliminación de un recurso, si el cambio afecta a la configuración perteneciente a un rol, el IWA presentará una pantalla a todos los usuarios on-line que estén desempeñando dicho rol. Desde esta pantalla se iniciará un debate donde cada usuario podrá expresar su opinión acerca de la incorporación o eliminación propuesta. El objetivo del debate es determinar conjuntamente si se realizará o no la modificación sobre la configuración de trabajo asociada al rol en cuestión.

En caso de acordarse la modificación, el IWA agregará o eliminará el recurso a la configuración del rol correspondiente, y actualizará los entornos de trabajo de todos los usuarios on-line que estén desempeñando el rol afectado por el cambio.

En el caso que la operación sea un alta o una baja no acordada, se guardará la discusión como parte del recurso, para que posteriormente pueda ser consultada por cualquier usuario interesado en la misma.



Pantalla 4.4 Selección de configuración afectada

4.6 DEBATE

Como ya mencionamos, en la incorporación o eliminación de un recurso a una configuración de trabajo, o en la incorporación o eliminación de un rol, se iniciará un debate entre los usuarios del grupo.

Dentro de este debate, los usuarios se identifican con uno de los siguientes roles:

- **Initiator:** es el rol tomado por el usuario que sugiere el cambio. Este usuario es el que inicia la conversación y el único con permiso para concluirla.
- **Participant:** el resto de los usuarios que deseen participar del debate.

Dependiendo del rol tomado en la discusión, un usuario podrá enviar distintos tipos de mensajes. El usuario que sugiere el cambio, automáticamente, tomará el rol *Initiator* y comenzará el debate mediante un mensaje de tipo **Proposition**. En este mensaje informará el cambio sugerido, explicando los beneficios del mismo.

El resto de los usuarios que quieran participar del debate, tomarán el rol *Participant* y podrán dar su opinión, aceptando o rechazando la propuesta recibida, mediante los mensajes de tipo **Acceptation** y **Negation**.

Durante la discusión, cualquier usuario, ya sea el iniciador de la misma o el resto de los participantes, podrán enviar mensajes de tipo **Question**, **Answer** y **Comment**, con el fin de obtener una comunicación más clara y ordenada.

El usuario iniciador, en el momento que crea conveniente, enviará un mensaje de tipo **Resolution** para informar la decisión tomada. Esta decisión la deberá tomar teniendo en cuenta la cantidad de mensajes recibidos de tipo aceptación y negación. Si la decisión es afirmativa, se reflejará el cambio en la configuración de trabajo correspondiente.

En cualquier momento, un usuario puede consultar y participar de los debates pendientes. Así como también, consultar las discusiones concluidas que se encuentran asociadas a una herramienta o rol, donde podrá observar quien tuvo la iniciativa del cambio, cuales fueron sus fundamentos y la opinión expresada por el resto de los participantes.

Con esta forma de trabajo en grupo, logramos que las personas trabajen de una manera más integrada, intercambiando experiencias personales, aprendiendo unas de otras y optimizando los tiempos de aprendizajes de los usuarios más novatos. Con la interacción planteada, negociación y discusión, se obtiene una comunicación que explota los aspectos de la colaboración.

4.7 EJEMPLO

Supongamos una empresa proveedora de software, donde trabajan personas que desempeñan uno o más de los siguientes roles: Analista, Programador y Tester. Cada rol necesita un conjunto específico de herramientas y documentos para llevar a cabo las tareas correspondientes.

Inicialmente, todos tendrían en su entorno de trabajo un conjunto básico de herramientas, compuesto por el Word, Excel, Outlook, Papelera, Maletín, Explorador de Windows. Además de esta configuración básica, el Analista podría tener el Project, el Programador el Eclipse, y el Tester el Internet Explorer y los programas que debería probar.

Así, cuando una persona debe realizar tareas de Analista, su entorno de trabajo se configuraría automáticamente con las herramientas correspondientes al rol (configuración básica + Project) Si en otro momento, esta misma persona debe realizar tareas de Tester, a su entorno de trabajo se agregaría el Internet Explorer y los programas que deben ser probados.

Lo interesante es que los usuarios que desempeñan el mismo rol aprenden unos de otros. A medida que un analista avanza en su tarea, puede descubrir nuevas formas de llevar a cabo su trabajo. Le puede surgir la necesidad de incorporar una nueva herramienta a su configuración, como por ejemplo, el editor gráfico Visio. Entonces, el analista evaluará si la nueva herramienta también es de utilidad para el resto de los usuarios que desempeñan el mismo rol o si sólo es útil para su trabajo específico. De esta manera, el analista tendrá la posibilidad de incorporarla a la configuración asociada a su identidad o podría tener lugar una comunicación con el resto de los usuarios que desembocaría en una nueva configuración “ideal” para analistas.

En este último caso, el analista explicaría cómo el uso de esta nueva herramienta los ayudaría en la realización de sus tareas cotidianas, generándose así un debate entre los usuarios que deseen participar de la discusión. Es posible que no se pongan de acuerdo y decidan que el cambio sugerido no sea realmente de “analista”, sino preferencia del usuario que sugirió el cambio. Con lo cual, decidirían dejarlo asociado a la identidad del usuario y no al rol de analista. En caso de ponerse de acuerdo, se incorporaría el Visio a la configuración del rol analista, evolucionando así de manera colaborativa a una nueva configuración “ideal” para dicho rol.

Más adelante, cuando otro analista entre al sistema, se le presentaría, automáticamente, la nueva configuración propuesta por su colega, teniendo la posibilidad de ver quien tuvo la iniciativa de incorporar la herramienta, cuales fueron sus motivos y la opinión de los usuarios que participaron en la negociación. De no estar de acuerdo, podría iniciar un nuevo debate para acordar la eliminación o no de la herramienta.

Por otro lado; un Programador, que además de realizar tareas de programación, documenta sus programas, puede necesitar la incorporación del rol “Documentador” al sistema. Con lo cual, de manera similar a la anterior, podría iniciar una nueva comunicación con el resto de los usuarios para negociar sobre la incorporación del nuevo rol.

CAPITULO 5 - IDEAL WORKSPACE

AGENT: IMPLEMENTACIÓN

5.1 ESTRUCTURA DE LA APLICACIÓN

Como mencionamos en el capítulo anterior, para desarrollar el IWA nos enfocamos en un entorno de trabajo utilizando Windows. La meta es extender la funcionalidad que provee dicho Sistema Operativo para el manejo de entornos de trabajo.

Dado que esta plataforma nos permite adaptar el ambiente de trabajo para un usuario y mantenerlo consistente en los sucesivos ingresos al mismo, nuestra herramienta deberá extender esta funcionalidad para cubrir los aspectos de colaboración y comunicación necesarios para poder obtener un entorno de trabajo compartido por los miembros de un grupo, que desarrollen la misma actividad, y que pueda evolucionar colaborativamente.

INTEGRACIÓN CON EL SISTEMA OPERATIVO

Al querer integrar el IWA con la plataforma elegida, surgió la necesidad de investigar el funcionamiento y la forma en que nuestra aplicación debía interactuar con dicho sistema operativo. Como producto de esta investigación decidimos utilizar un paquete de clases específico desarrollado por la compañía Servertec, la cual desarrolla productos y servicios de software innovadores destinados a desarrolladores de sitios Web.

El paquete, denominado Servertec Foundation Classes (SFC), está formado por un conjunto de clases y librerías nativas (dlls) que permiten el acceso a las funcionalidades nativas sin la necesidad de tener que escribir código en C/C++ o conocer el manejo interno de la plataforma. En particular, utilizamos este paquete para el manejo de los accesos directos, debido a que las librerías básicas provistas por Java no cubrían esta funcionalidad.

SOPORTE DE COLABORACIÓN

Para lograr la colaboración entre los miembros de un grupo, necesitábamos un modelo de datos compartido, es decir un medio o herramienta que nos permita:

- Guardar un **modelo de datos compartido y persistente** que se alimente de manera colaborativa por los usuarios del mismo.
- **Diversos modos de colaboración**, brindando soporte para la colaboración sincrónica (varios usuarios modificando el mismo objeto al mismo tiempo y capaz de observar los cambios que realizan los otros en el momento en que ocurren), así como también para la colaboración asincrónica (varios usuarios trabajando sobre un objeto común, cada uno, trabajando en un momento diferente y sólo capaz de ver los resultados de los cambios de los otros). Que también provea soporte para el uso mono-usuario y grupal, y para las transiciones entre éstas situaciones.
- **Independencia de plataforma**, para que permita dar soporte a los trabajadores de oficina móvil, que trabajan en diferentes situaciones y en plataformas variantes.

Por eso utilizamos DyCE, que es un framework realizado en Java para el desarrollo de componentes colaborativos, compuesto por una arquitectura Client-Server, que emplea una base de datos orientada a objetos que es replicada en los clientes. Además, sigue una metodología de desarrollo basada en componentes colaborativos, llamados Componentes Groupware. El IWA es un Componente DyCE. En el capítulo 6 hablaremos más en detalle sobre este framework.

SOPORTE DE COMUNICACIÓN

Por otro lado, durante el proceso de obtención de la configuración de trabajo “ideal”, debíamos lograr que los entornos de trabajo evolucionen de manera colaborativa para que las personas trabajen de una manera más integrada, intercambiando experiencias personales, aprendiendo unas de otras y optimizando los tiempos de aprendizajes de los usuarios más novatos.

Para lograr esto necesitábamos un medio de comunicación que permita la interacción, negociación y discusión de los miembros del grupo, como puede ser el intercambio de mensajes, el cual puede verse inicialmente como un “sistema de chat”.

Decimos inicialmente, porque un sistema de chat convencional no era suficiente para cubrir todos los aspectos de la negociación que planteamos en el capítulo anterior. Necesitábamos un tipo de chat que se pueda adaptar y que se integre con el resto del trabajo, es decir, que provea la posibilidad de:

- Definir distintos perfiles de usuarios dentro de un debate.
- Definir distintos tipos de mensajes para cada perfil.
- Estructurar los mensajes en forma jerárquica.
- Fácil integración con el resto del desarrollo.

Por las razones mencionadas, utilizamos Chatblocks para el desarrollo de la

comunicación. Que, como detallaremos en el próximo capítulo, es un framework de caja blanca desarrollado sobre DyCE que permite la construcción de herramientas de comunicación sincrónica basadas especialmente en texto.

Si bien chatblocks nos proveía una herramienta para implementar un chat adaptado a nuestras necesidades, fue necesario tratar la comunicación asincrónica desde el modelo compartido del IWA, ya que sólo provee comunicación sincrónica.

ESTRUCTURA DE LA APLICACIÓN

Con lo visto hasta el momento podemos mostrar gráficamente la estructura resultante de la aplicación:

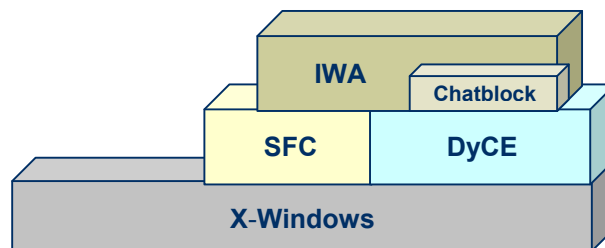


Gráfico 5.1 Estructura de la Aplicación

Resumiendo, utilizamos Windows como pilar principal de nuestro trabajo, la librería SFC para interactuar con dicho Sistema Operativo, el framework DyCE como soporte de la colaboración y Chatblocks como soporte de la comunicación.

5.2 MODELO GENÉRICO DE UNA CONFIGURACIÓN DE TRABAJO

Debido a que el IWA emplea DyCE para manejar la colaboración, y controlar la persistencia y distribución de los objetos compartidos, es una aplicación gobernada por las mismas reglas que son empleadas para generar nuevas componentes en DyCE: básicamente se compondrá de un modelo compartido (un ModelObject) y una interfaz local (el componente groupware, subclase de MobileComponent).

A continuación presentamos la arquitectura conceptual del modelo compartido del IWA, por motivos de claridad sólo incluimos las relaciones principales entre las clases. Para implementar este modelo compartido, diseñamos un modelo genérico

independiente de la plataforma de software, que permite modelar el concepto de entorno o configuración de trabajo de forma que pueda evolucionar de acuerdo a las necesidades de un grupo de trabajo. Este modelo puede extenderse a las distintas plataformas que tengan entorno gráfico, portales de Internet, etc.

En esta sección sólo describiremos las clases que conforman el modelo genérico.

Para representar el concepto de configuración de trabajo definimos la clase *WorkConfigurationModel*. Esta clase, está compuesta por una colección de objetos *WorkConfigurationItemModel*. Cada uno de ellos representa los distintos elementos que componen el entorno de trabajo. Cada uno de los elementos consta de una colección de Recursos representados por la clase abstracta *ResourceIWAModel*.

El *ResourceIWAModel* puede especializarse en un objeto *FileIWAModel*, *ShortcutIWAModel* o *FolderIWAModel* y mantiene una colección de *DebateModel*, compuesta por la discusión (negociación) que se realizó en el momento de su incorporación al entorno de trabajo; como también, todas las discusiones que involucren a dicho recurso.

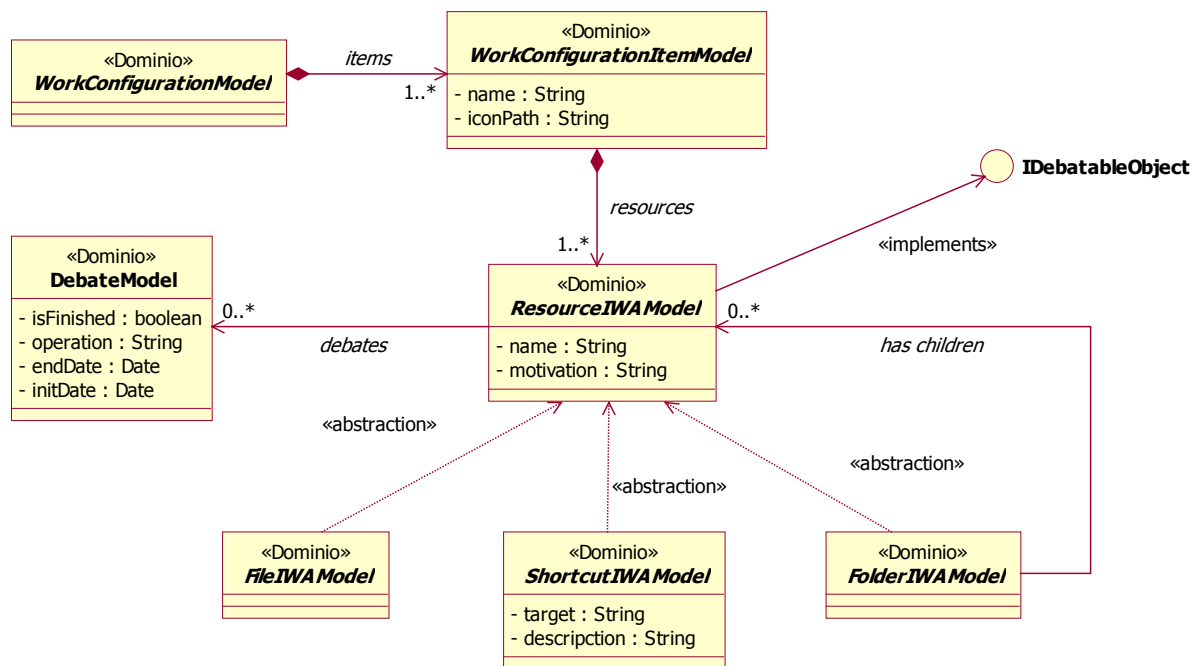


Diagrama 5.1. Modelo Genérico de una Configuración de Trabajo

5.3 EXTENSIÓN DEL MODELO GENÉRICO PARA EL CASO ELEGIDO

Para implementar el caso de estudio sobre la plataforma Windows 98, especializamos el modelo genérico, planteado en la sección anterior, para representar las características propias de esta plataforma.

Creamos la clase *WorkConfigurationWin* como subclase de *WorkConfigurationModel*. Esta especialización nos permite obtener un objeto que represente una configuración de trabajo con los elementos que componen un entorno de trabajo Windows, como pueden ser el Escritorio, Menú de Inicio, Favoritos, etc.

Para lograr esto, implementamos el método abstracto *loadItems()*, provisto por la clase *WorkConfigurationModel*. Cada especialización que se realice sobre el modelo genérico, deberá implementar dicho método para instanciar los elementos propios de cada plataforma.

De la misma forma, creamos la clase *WorkConfigurationItemWin* como especialización de *WorkConfigurationItemModel*, que es la encargada de implementar los métodos necesarios para el manejo del sistema de archivos de Windows.

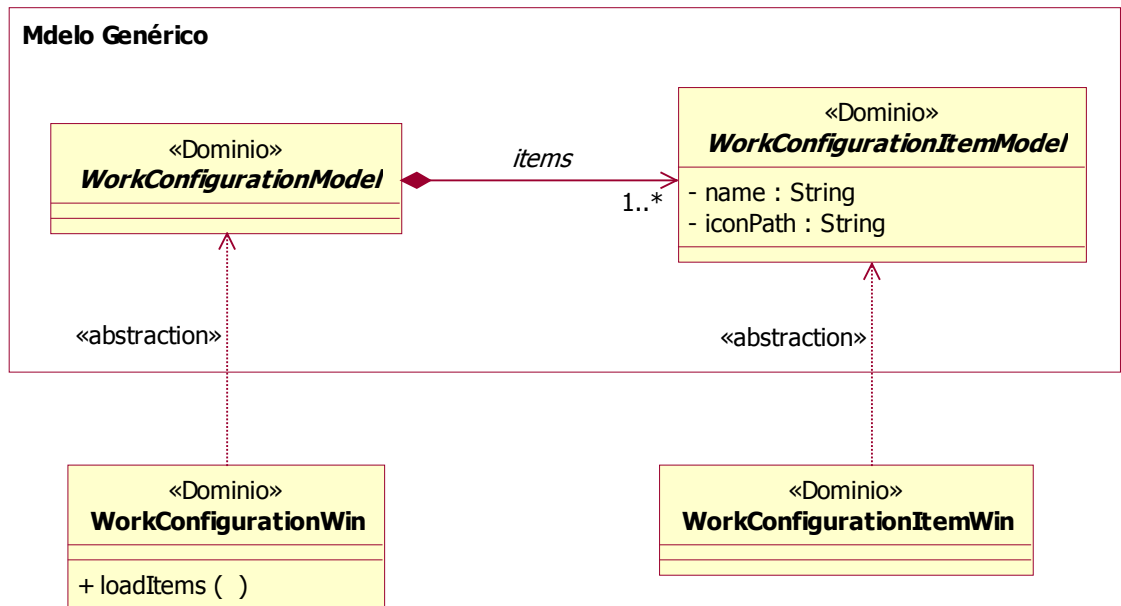


Diagrama 5.2. Extensión del Modelo Genérico para Windows (parte 1)

Por otro lado, creamos las clases *FileWin*, *ShortcutWin* y *FolderWin* como subclases de *FileIWAModel*, *ShortcutIWAModel* y *FolderIWAModel* respectivamente. Para cada una de estas subclases implementamos los métodos de lectura/escritura propios de la plataforma.

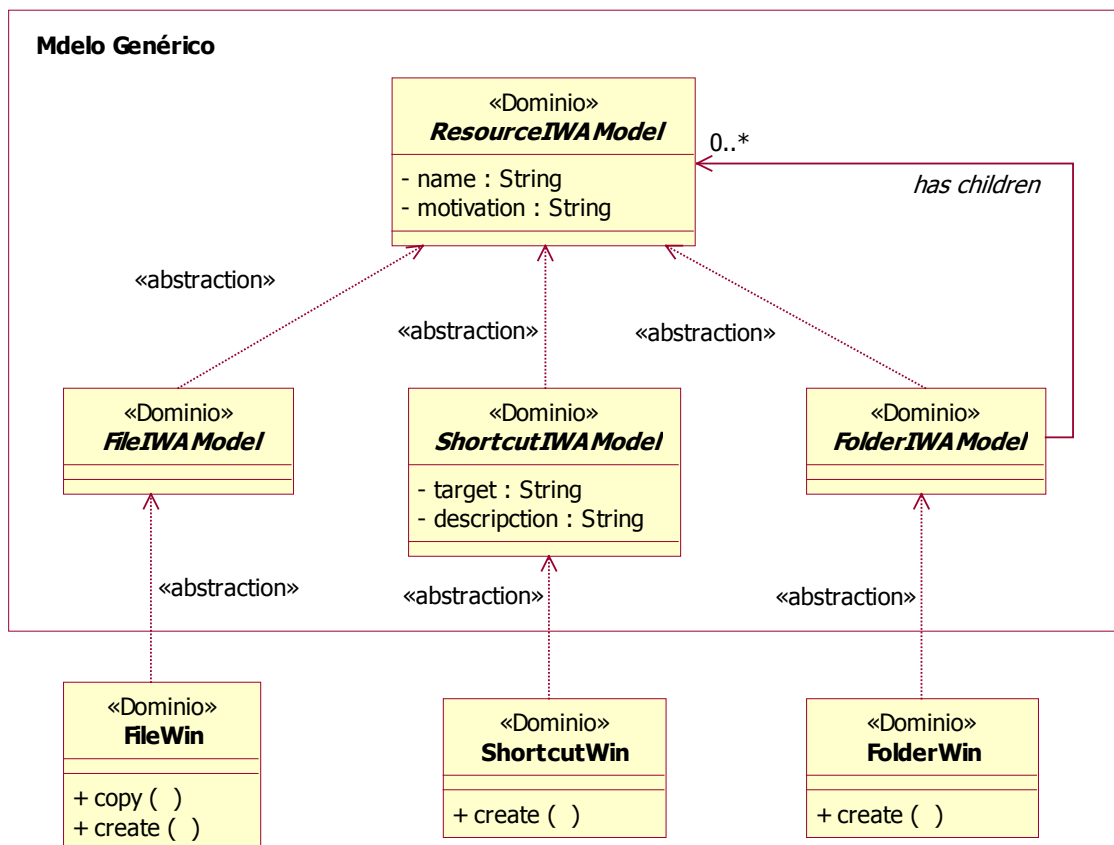


Diagrama 5.3. Extensión del Modelo Genérico para Windows (parte 2)

Con el modelo planteado logramos una fácil y rápida extensión del modelo genérico. Para implementarlo sobre una nueva plataforma sólo se requiere desarrollar las clases equivalentes a las mencionadas en esta sección e implementar los métodos abstractos que serán los encargados de realizar los accesos de lectura/escritura propios de la nueva plataforma.

5.4 IWA

Como mencionamos anteriormente, el IWA es un componente groupware de DyCE. Por tal motivo, todos los objetos del modelo que necesiten ser compartidos y persistentes deben extender de la clase *ModelObject* provista por DyCE.

Para que el IWA sea un componente groupware de DyCE, definimos la clase *IWAModel* como subclase de *ModelObject*.

En DyCE, cada vez que un cliente ingresa a uno de los componentes groupware que tiene publicado, se genera una nueva instancia del mismo. Dado que en nuestro caso, necesitábamos tener una única instancia del modelo de datos del IWA, debimos crear la clase *PersistentDataIWAModel*, que es la que contendrá los datos propios del dominio de nuestra aplicación.

De esta forma, cada vez que un usuario ingrese al IWA, se le generará una nueva instancia del *IWAModel*, y cuando sea necesario recuperar datos del modelo, lo hará dicha clase accediendo a los datos de la única instancia de la clase *PersistentDataIWAModel*.

Representamos los diferentes perfiles que pueden tomar los usuarios dentro del grupo de trabajo por medio de la clase *RoleModel*. Cada instancia de esta clase tendrá una instancia de la clase *WorkConfigurationModel*, que contendrá la configuración de trabajo que fue evolucionando colaborativamente para ese rol.

La clase *RoleModel* también guardará una colección de objetos *DebateModel*, donde cada uno representará alguna discusión/negociación realizada sobre la incorporación o intento de eliminación del rol.

Modelamos los usuarios del IWA por medio de la clase *UserModel*. Un usuario del IWA, esta formado por un usuario DyCE, y además contiene el comportamiento y funcionalidad necesarios para mantener y manejar su configuración de trabajo y los roles que fue desempeñando.

De la misma forma que la clase *RoleModel*, el usuario guardará una instancia de la clase *WorkConfigurationModel*, que contendrá la configuración de trabajo propia del usuario. También contendrá la historia de los distintos roles que fue desempeñando. Esto lo modelamos por medio de una colección de objetos *RolePeriodModel*, donde cada uno representa el período en el cual el usuario desempeñó un determinado rol dentro del grupo, contiene un *RoleModel* y las fechas que conforman dicho período.

Como la estructura de directorios y los programas instalados pueden variar de una máquina a otra, desarrollamos clase *LocalConfiguration*, que contiene información sobre la configuración de la máquina local de cada usuario.

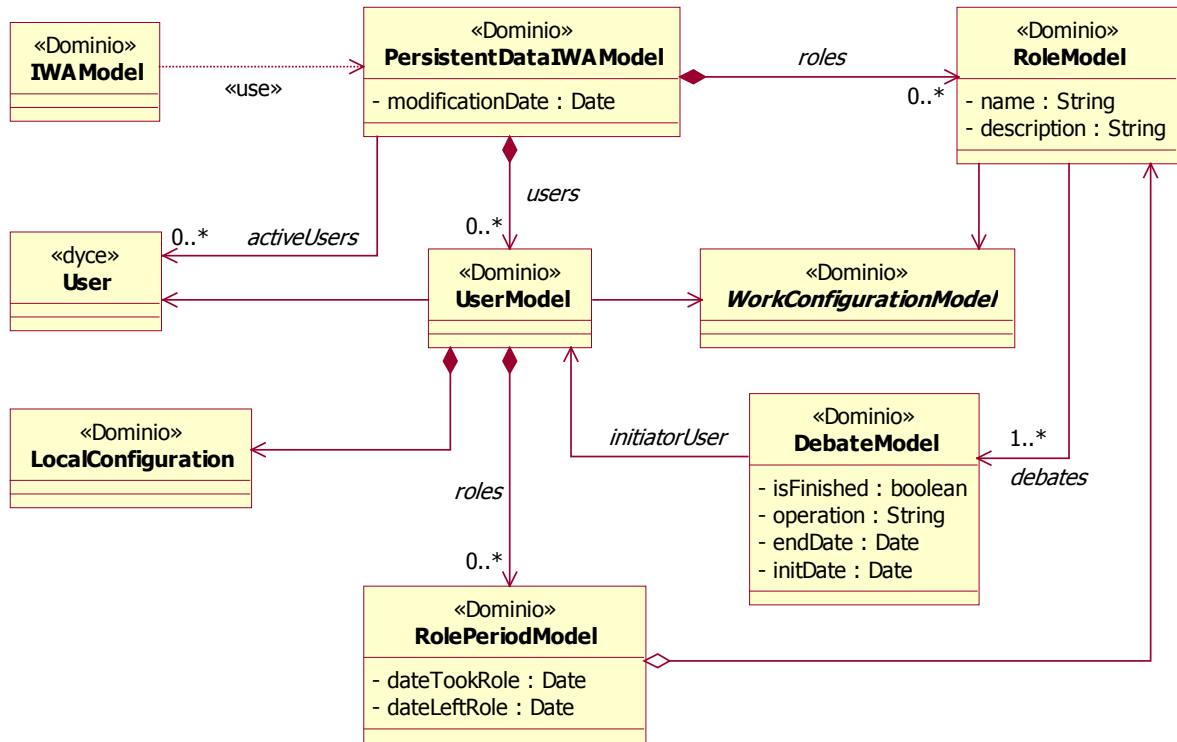


Diagrama 5.4 Modelo principal del IWA

Por lo tanto, el entorno de trabajo de un usuario está compuesto por la suma de las configuraciones de trabajo de los roles que ejerce y de la asociada a su identidad.

5.5 DEBATE

5.5.1 DECISIONES TOMADAS

Para implementar el debate entre los usuarios, estaba claro que lo haríamos mediante el uso de Chatblocks, pero debíamos analizar las alternativas posibles provistas por dicho framework, durante este análisis fuimos encontrando ventajas y desventajas para cada una de las alternativas que detallamos a continuación.

ALTERNATIVA 1: DISCUSIÓN USANDO UNA MODIFICACIÓN DE SNAPCHAT

Antes de comenzar a explicar esta alternativa, comentaremos brevemente de que se trata Snapchat para que se pueda comprender más fácilmente la alternativa planteada.

Snapchat

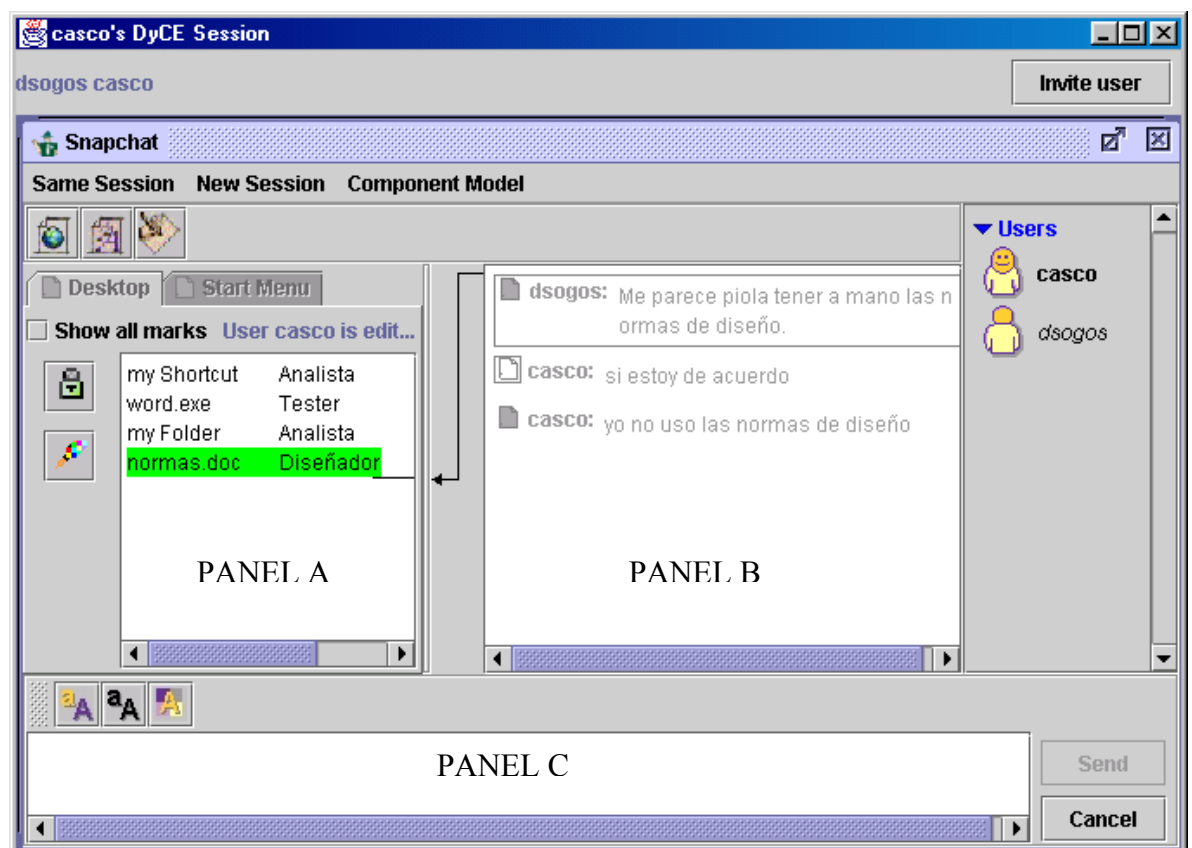
Snapchat es una herramienta creada con Chatblocks. Consiste de dos sistemas que interactúan entre sí: una aplicación chat y una pizarra colaborativa donde los participantes pueden mostrar y editar documentos.

La principal ventaja de tener asociada una herramienta de chat a la pizarra, radica en que se puede entablar una discusión acerca del contenido de algún documento mostrado en la pizarra.

La discusión se realiza creando *referencias* sobre alguna parte del documento. Es decir, cuando un usuario desea realizar algún comentario de alguna imagen o párrafo mostrado en el documento, simplemente lo selecciona y activa el comando marcar, con el que aparecerá un bloque donde podrá escribir y enviar un mensaje relacionado a la sección del documento marcada. El resto de los usuarios verá un nuevo mensaje, y bastará con seleccionarlo de la lista, para que puedan ver mediante un recuadro y una flecha a que zona del documento esta haciendo referencia.

Implementación usando Snapchat

En la Figura 6.1 se puede observar como quedaría una pantalla del debate utilizando Snapchat.



Pantalla 5.1 Implementación del Debate usando Snapchat

Para adaptar el funcionamiento del Snapchat actual al funcionamiento que deseamos para el IWA, debíamos realizar las siguientes modificaciones.

Se usaría siempre, el mismo snapchat para contener todas las discusiones pendientes (o sea no concluidas).

Los participantes de la discusión estarían identificados con los dos roles mencionados anteriormente:

- Initiator: usuario que sugiere el cambio. Inicia y concluye la conversación.
- Participant: todos los demás usuarios.

PANEL A

Cada pestaña sería un elemento (Item) de la Configuración de Trabajo, es decir, Desktop, StartMenu, etc. Las cuales aparecerían creadas al abrir el Snapchat.








Cada una de estas pestañas contendría una lista con los íconos o herramientas del Item, que estén involucrados en una discusión y que aún no se ha llegado a una decisión.

PANEL B

Los mensajes deberían poder mostrarse en estructura de árbol. Todos los mensajes relacionados con una herramienta colgarían de un mensaje de tipo **Proposition**. De esta forma se mostrarían las conversaciones pendientes de manera ordenada, sin mezclar mensajes pertenecientes a diferentes negociaciones.

PANEL C

Los mensajes tendrían los siguientes tipos:

	Tipo Mensaje	Descripción
	Proposition	Inicia la negociación, el usuario que sugiere el cambio explica los beneficios de llevar a cabo dicho cambio.
	Resolution	Cierra la negociación y se lleva a cabo lo acordado en el debate.
	Acceptation	Indica la aceptación del cambio sugerido.
	Negation	Indica el rechazo del cambio sugerido.
	Question	Pregunta sobre el cambio sugerido.
	Answer	Respuesta a una inquietud sobre el cambio sugerido.
	Comment	Comentario sobre el cambio sugerido.

El usuario que cumpla el rol “Initiator” podrá enviar mensajes de tipo: Proposition y Resolution. El resto de los usuarios que participen del debate, tendrán el rol “Participant” y podrán enviar mensajes de tipo: Acceptation y Negation.

Todos los usuarios podrán enviar mensajes de tipo: Question, Answer y Comment

Ejemplo

Desde el IWA un usuario podría querer agregar o eliminar el file xyz.doc del escritorio del rol Tester. Entonces, se abriría Snapchat:

- en el PANEL A, se agregará el elemento (xyz.doc Tester) en la pestaña Desktop.
- el PANEL C tendrá el foco, para que el usuario ingrese los motivos de su sugerencia.
- Luego, enviará el mensaje, cuyo tipo implícito será Proposition.
- En el PANEL B se agregará el mensaje Proposition, que quedará asociado al elemento recientemente agregado.

Para el resto de los usuarios on-line, podríamos forzar la apertura del snapchat, para que puedan participar de la discusión, para esto:

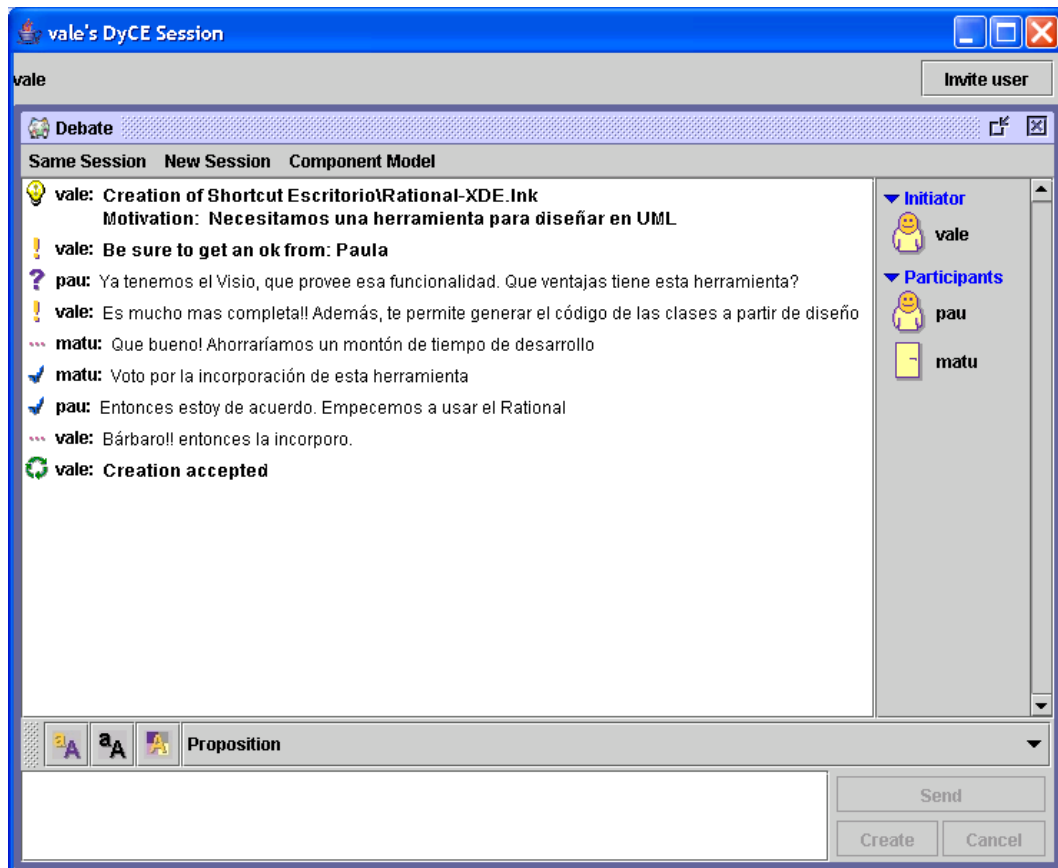
- el usuario interesado en dar su opinión debería seleccionar el elemento (en el PANEL A) que originó la discusión.
- Se desplegaría un combo en el PANEL C, con los tipos de mensajes que puede enviar.
- El mensaje enviado se ubicaría (en el PANEL C) en estructura de árbol, como un hijo del mensaje de tipo Proposition enviado por el Initiator.

De esta manera, los usuarios darían su opinión, aceptarían o rechazarían la propuesta. Teniendo en cuenta la cantidad de mensajes de tipo Acceptation y Negation, el Initiator, en el momento que crea conveniente, enviaría un mensaje de tipo Resolution para informar la decisión tomada. Si la decisión es afirmativa, se reflejará el cambio en la configuración de trabajo afectada.

En cualquier momento, los usuarios podrían abrir el snapchat para ver y/o participar de las negociaciones pendientes.

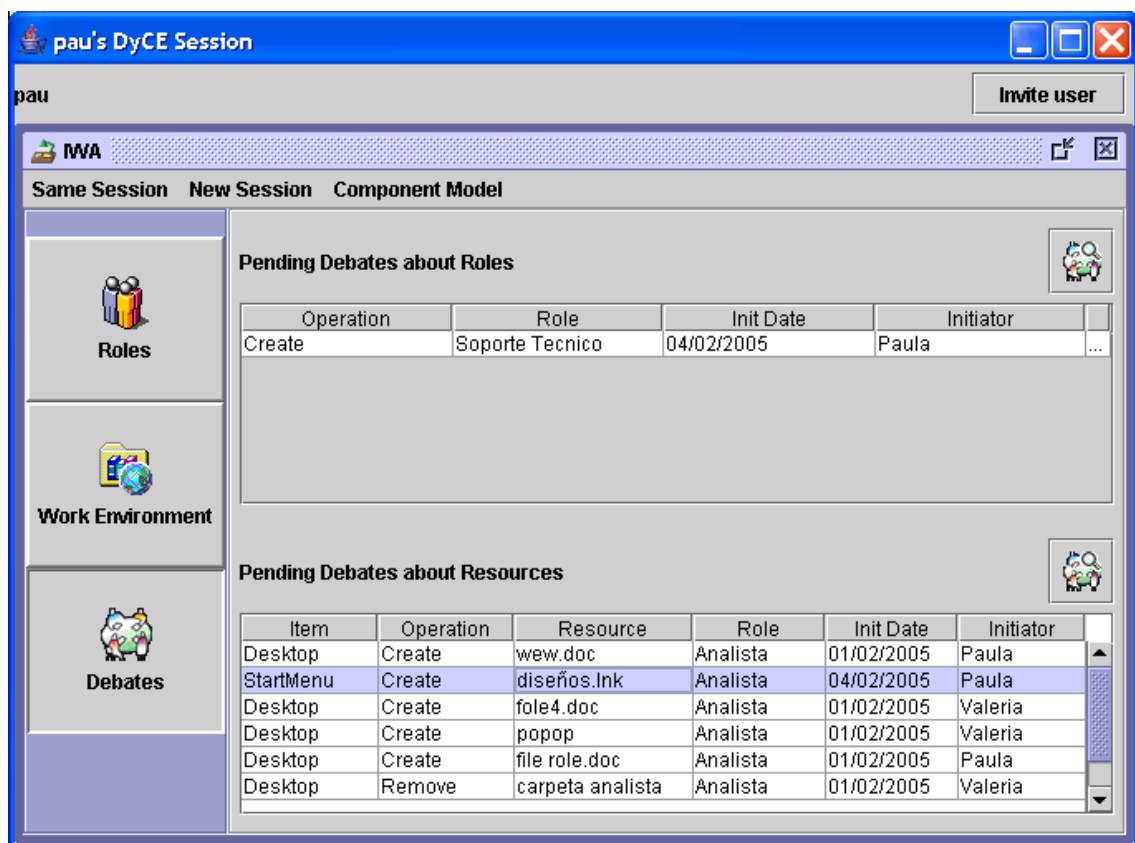
ALTERNATIVA 2: DISCUSIÓN SIN USAR SNAPCHAT

En lugar de snapchat, se utilizaría una combinación del resto de los chats, que solo contendría los PANELES B y C.



Pantalla 5.2 Debate

El PANEL A se reemplazaría, en el IWA, por una lista de herramientas con discusiones pendientes. En cualquier momento, los usuarios tendrían la posibilidad de seleccionar una herramienta y abrir su discusión asociada. De esta forma, en el PANEL B solo se vería la discusión de la herramienta seleccionada, donde el usuario podrá participar de la misma.



Pantalla 5.3 Debate pendientes

Al igual que en la alternativa anterior, un usuario podrá participar de varias negociaciones al mismo tiempo, con la diferencia de que esto lo haría desde distintas ventanas de discusión.

El resto de la operatoria seguiría siendo la misma (roles, tipos de mensajes, etc.)

ALTERNATIVA IMPLEMENTADA

El inconveniente que encontramos con la alternativa que utiliza Snapchat, es que se presentan todos los debates pendientes en una misma ventana, y esto puede resultar poco claro y confundir a los miembros del grupo, dado que los usuarios cambian de rol entre un debate y otro, según la herramienta sobre la que se esté discutiendo.

Por ejemplo, supongamos que se visualizan tres debates pendientes, de los cuales el usuario1 solo propuso el cambio sobre la herramienta1, con lo cual, cuando este usuario seleccione dicha discusión tomará el rol Initiator; mientras que cuando seleccione alguna de las otras dos discusiones tendrá el rol Participant.

Además, en el caso de que un usuario este participando de varios debates a la vez, lo

estaría haciendo sobre la misma ventana de chat lo que puede resultarle confuso y hacerle perder la noción sobre qué se está discutiendo.

En cambio, con la segunda alternativa, vemos que se simplifica la operatoria de los usuarios, ya que, a pesar de que podrán participar de varios debates al mismo tiempo, lo harán desde diferentes ventanas, lo que les permitirá concentrarse en la negociación de su interés.

Además, cuenta con una ventaja adicional sobre la primera alternativa. Los mensajes que componen el debate, se presentarán en forma de árbol, con la característica de que el usuario podrá seleccionar un mensaje específico al que quiere hacer referencia; quedando así una conversación estructurada que se visualizará como un árbol con mas de un nivel.

Por todo lo expuesto en esta sección, decidimos implementar la negociación/discusión utilizando la alternativa 2, es decir, sin utilizar Snapchat.

5.5.2 MODELO

Para representar el debate desarrollamos la clase *DebateModel*. Esta clase extiende de *ChatApplication* para heredar toda la funcionalidad provista por chatblocks.

Dado que en el IWA manejamos debates sincrónicos y asincrónicos, y que chatblocks sólo provee la posibilidad de una comunicación sincrónica, para la implementación de la comunicación asincrónica incorporamos en el *DebateModel* la fecha de inicio de la negociación/discusión y la fecha de conclusión del mismo, que es la que nos indicará si el debate aún está pendiente.

Al mismo tiempo debimos proveer la manera de guardar y acceder a esos debates pendientes para brindarles a los usuarios la posibilidad de reanudarlos en cualquier momento. Para esto creamos la clase *PendingDebatesModel*, que contiene dos colecciones, una para guardar los debates no concluidos generados para debatir sobre la incorporación o eliminación de algún recurso a una configuración de trabajo, y otra colección para guardar los debates no concluidos generados para debatir sobre la incorporación o eliminación de algún rol al IWA.

Cada una de estas colecciones está representada por un *HashMap*, donde la clave es el debate pendiente y el valor asociado a dicha clave es un objeto *DebateDataModel*. Éste contiene el resto de la información necesaria para llevar a cabo la inserción o eliminación del objeto sobre el que se está debatiendo; por ejemplo, en el caso que el objeto sea un recurso, será necesario mantener información sobre el ítem de la configuración de trabajo sobre la que se quiere realizar la operación, como también el rol al que pertenece dicha configuración.

El IWA también mantiene los debates concluidos para ser consultados por cualquier usuario en cualquier momento. Por lo tanto, al concluir un debate se elimina de la

colección de debates pendientes y se agrega a la lista de debates asociada al objeto sobre el que se realizó la discusión, es decir, a la lista de debates de un *RoleModel* o de un *ResourceIWAModel*.

Por otro lado, definimos la clase *DebateConfiguration*, que es la encargada de crear una nueva discusión/negociación. Esto implica crear una instancia del *DebateModel*, incluirlo en la colección de debates pendientes y agregarle los usuarios que van a participar en el mismo.

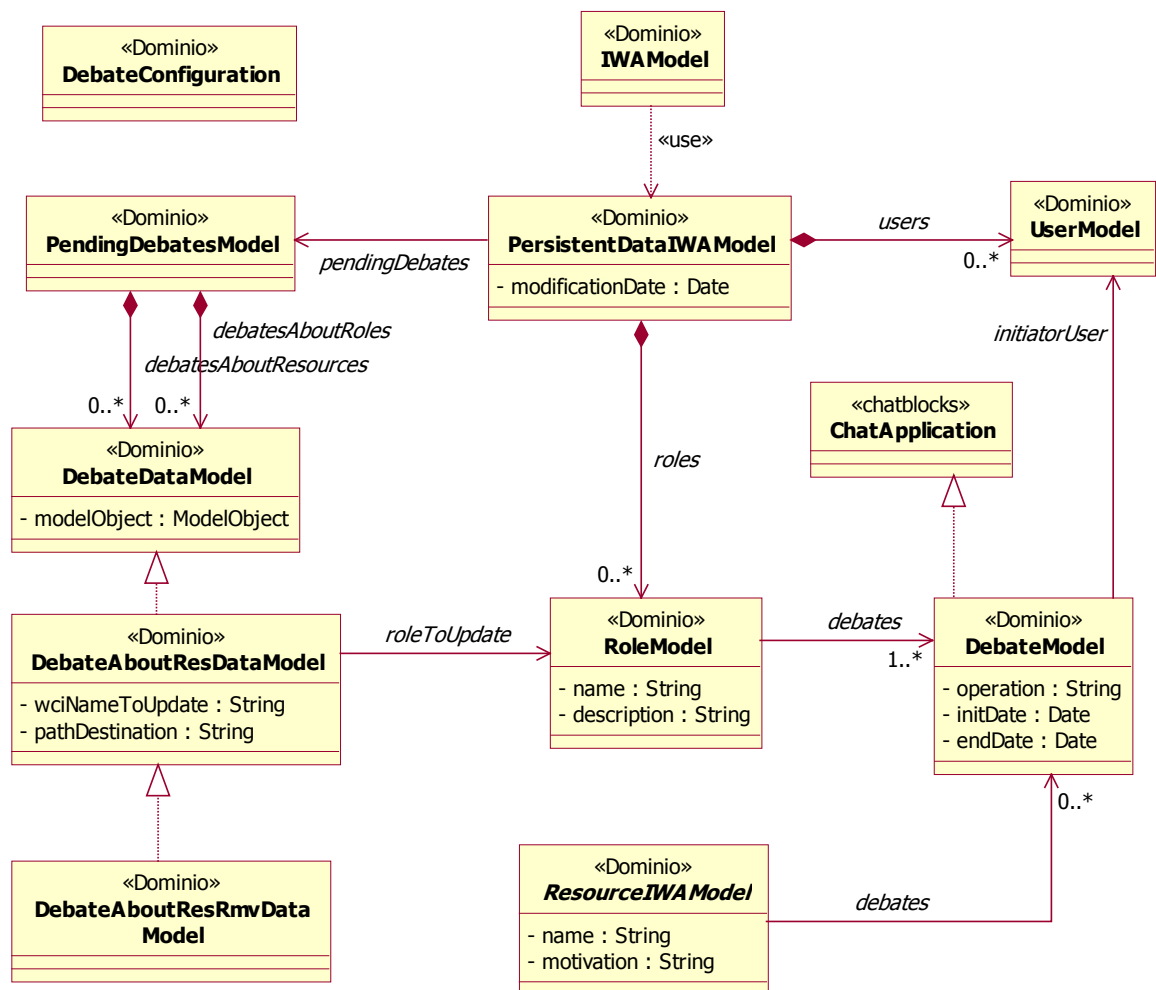


Diagrama 5.5. Modelo del Debate

5.6 CONFIGURACIÓN LOCAL

Cuando hablamos de Configuración Local nos referimos a la estructura de directorios y a la ubicación física de los recursos que componen el entorno de trabajo de cada usuario.

Si bien el entorno de trabajo de un usuario va evolucionando a medida que evolucionan las configuraciones de trabajo que lo componen, y que estas configuraciones son compartidas por los diferentes usuarios, la ubicación física de cada recurso difiere de un usuario a otro.

Por tal motivo, cada usuario deberá mantener su configuración local, que estará representada por una instancia del objeto *LocalConfiguration*.

Cada vez que un usuario ingrese al IWA, se deberá actualizar su configuración local con la información obtenida del sistema de archivos de dicho usuario.

La clase *LocalConfiguration* sólo contendrá el path del destino de cada shortcut, que se irá actualizando a medida que su entorno de trabajo evolucione. No es necesario guardar la ubicación física de los folders y files, dado que los mismos se encuentran ubicados sobre alguno de los ítems que componen una configuración de trabajo, y la ubicación física de estos es obtenida dinámicamente de la plataforma.

Para representar la información de los shortcuts, definimos la clase *LocalShortcut*, que contiene el nombre y el path destino al que apunta dicho shortcut.

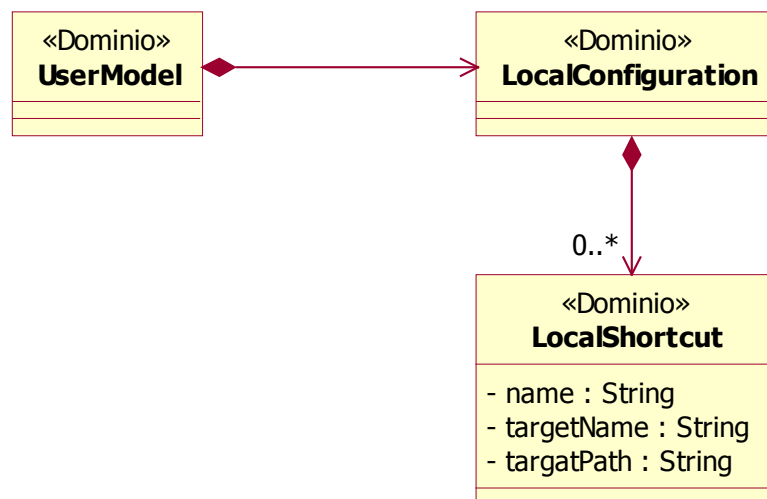


Diagrama 5.6. Modelo de la configuración local del usuario

Cuando un usuario ingresa al IWA, la aplicación se comporta de distintas formas, dependiendo de si es un usuario nuevo o si es uno existente:

- Si es un usuario nuevo, se crea una instancia del objeto *LocalConfiguration* para dicho usuario con los objetos *LocalShortcuts* correspondientes a cada uno de los links que en ese momento se encuentren en su entorno de trabajo.
- Si es un usuario existente, se recuperará la instancia creada del *LocalConfiguration*. Cuando el IWA actualice su entorno de trabajo en base a las configuraciones que le correspondan al usuario, utilizará la instancia de la *LocalConfiguration* recuperada para poder crear los links incluidos en su entorno de trabajo.

Cuando un usuario desee crear un shortcut deberá elegir a que configuración de trabajo desea agregarlo:

- Si decide agregarlo a la configuración asociada a su identidad, se creará un objeto *LocalShortcut* con los datos del nuevo recurso, que se agregará a su instancia del objeto *LocalConfiguration*.
- En cambio, si decide agregarlo en la configuración de alguno de los roles que está desempeñando, luego de debatir con el resto de los integrantes del grupo, se creará un objeto *LocalShortcut* con los datos del nuevo recurso en su configuración local, y cuando se deba actualizar el entorno de trabajo del resto de los usuarios que desempeñen el mismo rol, se hará una búsqueda para ubicar los datos del nuevo recurso en la sistema de archivos local, y así agregarlo a la instancia de la *LocalConfiguration* asociada a dichos usuarios.

CAPITULO 6 - DYCE

En este capítulo desarrollamos todo lo referente a la filosofía del framework DyCE (**D**ynamic **C**ollaboration **E**nvironment), como permite la generación de componentes y las ideas principales ligadas a la construcción de un componente groupware. El manejo de sesiones y la arquitectura cliente-servidor que DyCE plantea son mencionado sin dar demasiados detalles de implementación, solo lo necesario para comprender como es el desarrollo en DyCE, y en que forma fue pensado y diseñado nuestro trabajo.

DyCE es un framework realizado en Java para el desarrollo de componentes colaborativos, llamados Componentes Groupware. Dichos componentes pueden distribuirse por la red y ser combinados con otros componentes para formar nuevos ambientes de soporte de colaboración. Este framework está basado en la replicación dinámica de objetos, código móvil y un modelo de programación para desarrollar nuevos componentes.

6.1 OBJETIVOS Y REQUERIMIENTOS

DyCE cumple con una serie de requerimientos necesarios para soportar las exigencias de reusabilidad y flexibilidad.

Combinación y reuso de herramientas

Para soportar un amplio rango de diferentes situaciones colaborativas, los usuarios deben poder colaborar usando varias y diferentes herramientas colaborativas. Las herramientas de diferentes tipos necesitan ser acopladas en el mismo objeto compartido de trabajo colaborativo. Esta necesidad surge claramente en varios ambientes de colaboración, en los cuales el conjunto de colaboradores, sus capacidades en el sistema y las herramientas que tienen a su disposición no son homogéneos; por ejemplo, en un ambiente de enseñanza, donde el profesor tiene un conjunto de capacidades y herramientas disponibles diferente a la de los estudiantes.

Adicionalmente, las herramientas colaborativas necesitan ser usadas en diferentes contextos de trabajo (diferentes equipos, diferentes tareas cooperativas) y en combinación con otras herramientas.

Extensibilidad

El ambiente de groupware usado para las tareas colaborativas necesita ser extensible y customizable en tiempo de ejecución.

La customización puede darse en dos formas:

- Customización del programador, es decir, que los programadores experimentados pueden desarrollar nuevas herramientas para el sistema, y
- Customización de usuario final, es decir, que los usuarios finales tienen la habilidad extender y adaptar el sistema según sus necesidades.

Relacionado con la extensibilidad se encuentra el requisito de fácil publicación de las nuevas herramientas de groupware en tiempo de ejecución (es decir, sin tener que parar el ambiente de colaboración para que las nuevas herramientas estén disponibles). Este requisito se vuelve el más importante de todos a medida que el ambiente de groupware crece y cambia para adaptarse a las necesidades cambiantes de los usuarios.

Independencia de plataforma

Para dar soporte a los trabajadores de oficina móvil, que trabajan en diferentes situaciones y en plataformas variantes, el sistema necesita ser tan independiente de la plataforma como sea posible. Una plataforma de entrega muy importante en este contexto es la de Internet/Intranet. Debido a su amplia disponibilidad, esta plataforma posibilita la colaboración distribuida (tal como se requeriría en las empresas virtuales o extendidas y en otros equipos globalmente distribuidos, como por ejemplo los soportados por intranets corporativas).

Diversos modos de colaboración

Con el fin de ser usadas en un amplio rango de escenas, las herramientas usadas en el ambiente deben proveer soporte para la colaboración sincrónica (varios usuarios modificando el mismo objeto al mismo tiempo y capaz de observar los cambios que realizan los otros en el momento en que ocurren), así como también debe dar soporte para la colaboración asincrónica (varios usuarios trabajando sobre un objeto común, cada uno, trabajando en un momento diferente y sólo capaz de ver los resultados de los cambios de los otros). También debe proveer soporte para el uso mono-usuario y grupal, y para las transiciones entre éstas situaciones.

6.2 DISEÑO DEL SISTEMA

DyCE es un framework para desarrollo de groupware ideado para cubrir los requerimientos detallados en la sección anterior. Sirve como base para el desarrollo de nuevas herramientas de groupware y provee un ambiente para el uso de las mismas.

DyCE sigue una metodología de desarrollo basada en componentes, ya que esta metodología es adecuada para soportar los requerimientos de flexibilidad y extensibilidad enunciados anteriormente.

6.2.1 COMPONENTES GROUPWARE

Antes de seguir avanzando en la descripción del DyCE, es necesario definir el concepto central usado en el framework: *Componentes Groupware*.

- Los Componentes Groupware son herramientas visualmente interactivas y potencialmente complejas que permiten que múltiples usuarios accedan y manipulen datos compartidos (el modelo de dominio de la aplicación colaborativa)
- Los Componentes Groupware, típicamente, son más complejos que los componentes de una interfase monousuario, como puede ser un elemento de ingreso de texto. Los ejemplos de Componentes Groupware pueden ser un editor de dibujo compartido o un elemento de awareness visual. Estos componentes pueden reusarse y combinarse con otros componentes para formar nuevos ambientes de la colaboración.
- Los Componentes Groupware son conscientes de la colaboración, es decir son elementos de interacción que pueden acceder a información sobre la situación actual de la colaboración; por ejemplo, para adaptar su display según cada caso, o para proveer conocimiento del grupo a los usuarios.

6.2.2 ARQUITECTURA DEL SISTEMA

DyCE es un sistema de groupware distribuido Cliente-Servidor, desarrollado en Java. En la Figura 5.1 mostramos la arquitectura general de DyCE.

La comunicación Cliente-Servidor se hace usando una combinación de Java RMI (Remote Method Invocation). De esta manera, DyCE puede ejecutarse en ambientes donde se encuentre disponible un ambiente de ejecución Java. Actualmente, los ambientes Java también están disponibles en los dispositivos portátiles y móviles, por ejemplo, en la forma de Java 2 Edición de Micro, J2ME.

DyCE provee un Escritorio Groupware, desde el cual los usuarios pueden iniciar sesiones colaborativas, acceder a Componentes Groupware y a objetos del dominio.

En el sistema DyCE distribuido, el servidor tiene varias responsabilidades:

Object Management

El Object Manager del servidor mantiene todos los objetos compartidos usados en las colaboraciones. Se usan las interfases RMI de acceso y consulta para recuperar los objetos del Object Manager. Los objetos compartidos son almacenados

persistentemente en una base de datos de objetos para permitir la persistencia de los cambios, como también la colaboración asincrónica.

Transaction Management

Los clientes cambian los objetos compartidos de una manera transaccionalmente segura. Los accesos y modificaciones a los objetos compartidos son encapsulados en transacciones, y validados por el servidor. Para hacer que los Componentes Groupware sean más sensitivos, las transacciones son procesadas de forma completamente optimista y, cuando se detectan conflictos, son selectivamente canceladas.

Component Management

Los Componentes Groupware disponibles en el sistema son manejados por el Component Broker y almacenados en el Component Repository del servidor. Cuando un cliente requiere un componente, se consulta el conjunto de componentes registrados en el Component Broker para hallar el componente correcto (y localizar dónde esta almacenada su implementación).

Se pueden registrar los nuevos componentes para que estén disponibles a todos los usuarios.

Session Management

El servidor de DyCE administra todas las sesiones colaborativas que están activas en el sistema. Cuando un nuevo cliente se une a una sesión, se recupera la información de la sesión desde el Session Manager del servidor.

HTTP Service

El servidor de DyCE incluye su propio servicio HTTP, que es usado cuando se transfieren las implementaciones de los Componentes Groupware. Una vez que un componente requerido es identificado por el Component Broker, se descarga dinámicamente en el cliente usando HTTP. Los nuevos componentes, que los diseñadores desean publicar en un sistema DyCE que ya está en funcionamiento, también pueden ser subidos al Component Repository de la misma manera. DyCE utiliza el apoyo de Java por transmitir el código dinámicamente vía una conexión HTTP.

El servidor HTTP extensible incluido en DyCE, también puede usarse para manejar sitios de internet o intranet (o partes de ellos), que también pueden contener elementos de información de los objetos compartido y de las sesiones colaborativas como parte de su contenido Web.

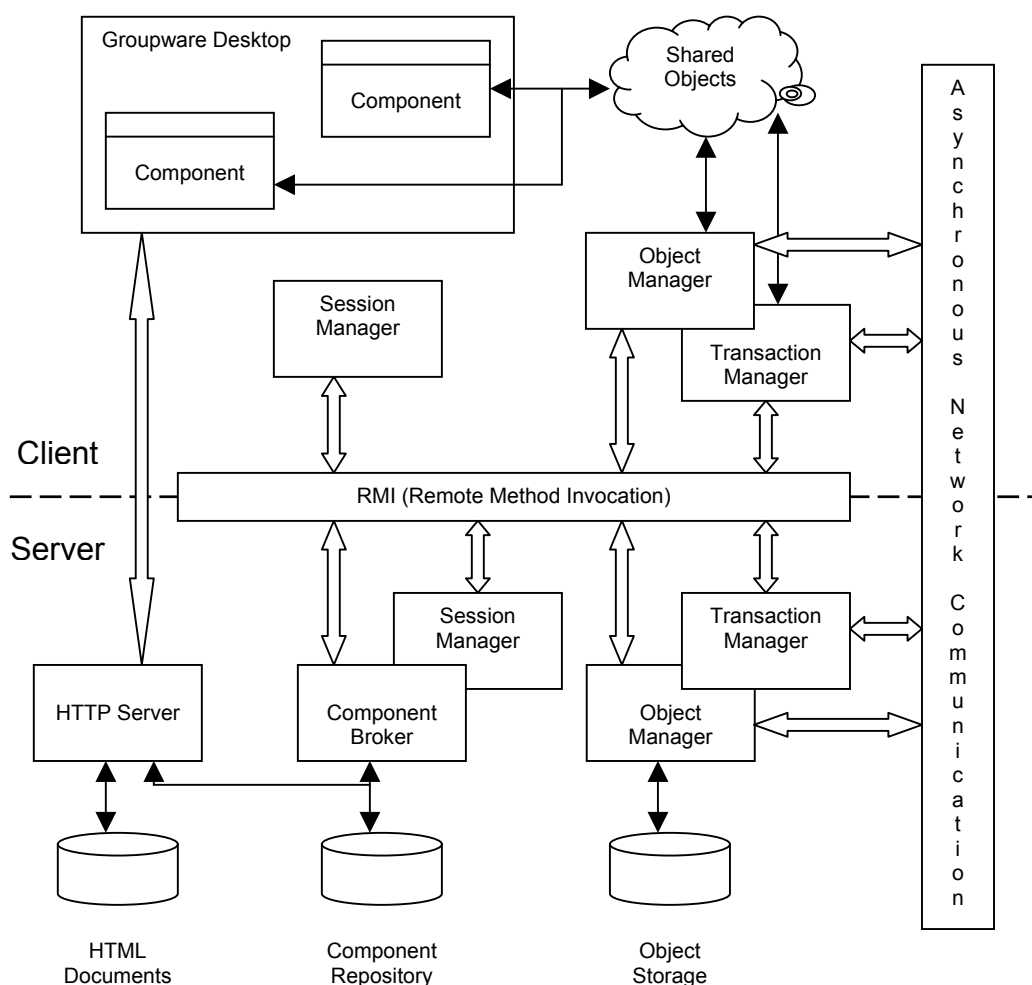


Figura 6.1 - Arquitectura del Sistema DyCE

6.2.3 OBJETOS COMPARTIDOS

Las herramientas colaborativas, tales como los Componentes Groupware, pueden acoplarse para proveer a un grupo de usuarios la idea de estar interactuando con un estado compartido común.

Se pueden identificar tres metodologías básicas para realizar el acoplamiento en los ambientes colaborativos, cada uno adecuado para diferentes requisitos. En una *aplicación compartida* (o ventana compartida) la arquitectura y la salida de una aplicación que se ejecuta en forma centralizada (típicamente monousuario) se distribuyen a todas las workstations de los usuarios. Las entradas de los usuarios son capturadas y transmitidas a la aplicación central en forma serializada, la cual reacciona como si estuviera siendo controlada por un solo usuario. La ventaja de usar casi cualquier aplicación monousuario colaborativamente es pagada con una falta de flexibilidad en la colaboración: sólo puede soportarse WYSIWIS estricto y colaboración

sincrónica. Además, las metodologías de ventanas compartidas, típicamente, tienen altos requisitos de ancho de banda.

La segunda metodología está dada por los eventos GUI compartidos. Aquí, cada usuario ejecuta una instancia de la aplicación compartida. Los eventos GUI de la aplicación son capturados, distribuidos y propagados a cada instancia de la aplicación como si se hubieran realizados localmente. Esta metodología sólo permite el acoplamiento de aplicaciones exactamente similares y también requiere WYSIWIS estricto para que los eventos GUI distribuidos realicen operaciones similares en todas las máquinas conectadas. La ventaja que presenta es que requiere menos ancho de banda que la metodología de ventana compartida.

La tercer metodología se basa en objetos compartidos, donde cada usuario tiene acceso a una instancia de la aplicación, la cual provee acceso a un conjunto de objetos de datos comunes. Estos objetos de datos pueden estar centralizados (es decir que sólo existe una copia de ellos) o pueden estar replicados (es decir que existen múltiples copias que se guardan en un estado consistente). Esta metodología, a veces conocida como MVC distribuido (Model/View/Controller), permite la más alta flexibilidad en el ambiente de colaboración. Si se modifican los objetos compartidos de una manera semánticamente consistente, se pueden acoplar diferentes herramientas usando un modelo compartido. Además, es posible lograr diferentes modos de acoplamiento y diferentes combinaciones de colaboración sincrónica y asincrónica.

Debido a su alto grado de flexibilidad, la metodología de acoplamiento implementada en DyCE es la de objetos compartidos: los datos compartidos están separados de los componentes usados para interactuar con tales datos compartidos, en un modo similar a la separación de aplicaciones y modelos del dominio.

El estado compartido es modelado como un conjunto de objetos compartidos, los cuales son replicados dinámicamente en el sistema distribuido, modificados por los Componentes de una manera transaccionalmente segura y mantenidos en un estado globalmente consistente. Cada cliente posee una réplica local de los objetos que son actualmente requeridos (porque están siendo visualizados o manipulados por Componentes Groupware usados en el cliente).

Los objetos compartidos son modelados como el modelo de datos de los Componentes Groupware (o modelo de dominio), el cual provee interfaces para acceder y modificar el estado del modelo de datos compartido. Un mecanismo observador une el/los componente(s) al modelo de datos compartido y permite cambiar notificaciones para que sean propagadas a los componentes correspondientes, los cuales pueden, consecuentemente, actualizar sus vistas.

Los objetos de los datos compartidos se guardan persistentemente en el servidor DyCE y son replicados dinámicamente a los clientes cuando son requerido por un componente.

El framework DyCE provee las clases básicas para desarrollo de los Componentes Groupware, así como los objetos del modelo usados por ellos, *MobileComponent* y *ModelObject*, respectivamente.

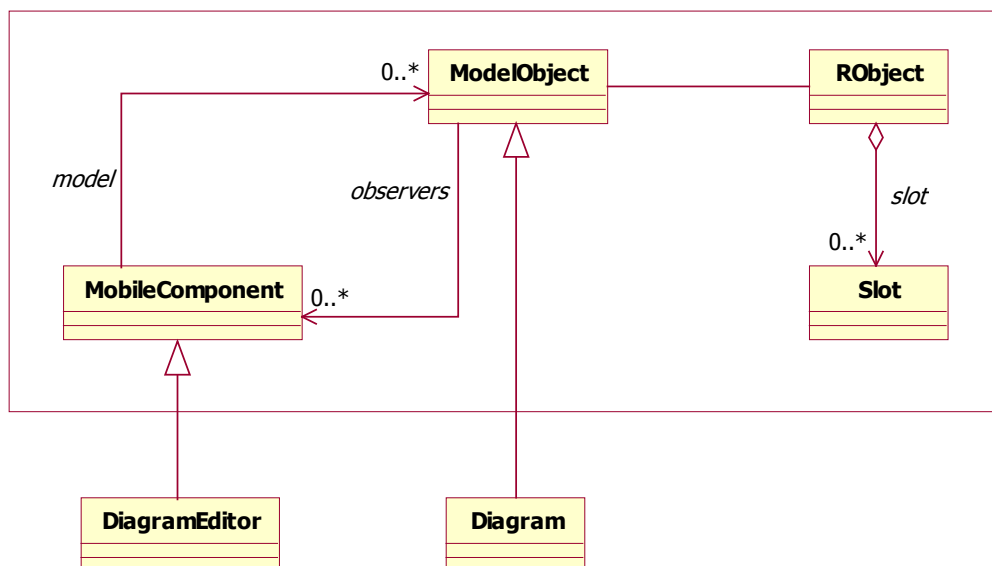


Figura 6.2 - Component and model hierarchy

Las clases derivadas de *ModelObject* proporcionan una interfase semánticamente significativa a los elementos de datos generales, los cuales están contenidos en un conjunto homogéneo de instancias de *RObject*. Las subclases de *ModelObject* definen el esquema de objetos compartidos en un modelo orientado a objetos, de manera similar a la definición de un esquema de base de datos. De esta manera, el mecanismo de replicación sólo necesita manejar instancias de *RObject*, ya que las instancias de *ModelObject* relacionadas pueden ser recreadas localmente. Las subclases de *ModelObject* también mantienen lógica de aplicación específica del dominio directamente asociada con los datos compartidos, y no específica a un cierto componente. De esta manera, se tiene una triple separación: Las clases de los componentes usadas para mostrar e interactuar con los objetos compartidos, las clases del modelo de dominio, que proveen una interfase, semánticamente significativa, específica del dominio a los datos compartidos, y los datos compartidos reales, que están contenidos en un conjunto de objetos homogéneo dinámicamente replicado. Esta separación es muy útil y robusta para el desarrollo de nuevos componentes. Los *RObjects* son almacenados persistentemente en el servidor. Siempre que el conjunto de elementos de datos permanezca estable, se pueden realizar cambios y modificaciones en las implementaciones de las subclases de *ModelObject* y *MobileComponent* y el esquema de *RObjects* continuará siendo válido.

En el diagrama de la Figura 6.2, la línea punteada muestra el límite entre el desarrollo

del framework y la extensión del mismo. El componente *DiagramEditor* deriva de la clase *MobileComponent* y los datos del diagrama se guardan en la clase *Diagram*, que deriva de *ModelObject*. En tiempo de ejecución, cuando se crean las instancias de la clase *Diagram*, se crea una instancia de *RObject* asociada, para guardar los datos compartidos replicados (en los Slots).

6.2.4 MOVILIDAD DE COMPONENTES (COMPONENT MOBILITY)

Un requisito importante de DyCE es el soporte de la extensibilidad en tiempo de ejecución. Esta extensibilidad se refiere a que se puedan agregar nuevos componentes mientras el sistema se esté ejecutando, y que estén inmediatamente disponibles a los usuarios.

Para dar soporte a este requerimiento, se hace uso de la habilidad que tiene Java para descargar dinámicamente implementaciones de clases desde el servidor, instanciar estas clases y ejecutar su código (también llamado *code mobility*). Los nuevos componentes pueden cargarse en el Component Repository usando el servicio HTTP y automáticamente serán agregados al Component Broker.

Este soporte permite proveer a los usuarios de una aplicación de inicio simple, que puede ser extendida en base a la situación de cooperación actual. La información sobre los componentes disponibles es modelada como un modelo compartido de DyCE y, por lo tanto, es guardada automáticamente en un estado consistente en todos los clientes conectados. Cuando un usuario accede a esta información y abre los nuevos componentes, estos son descargados en su máquina y abiertos localmente.

Las implementaciones del modelo de objetos asociado son descargadas junto con las implementaciones de los componentes. Las nuevas instancias creadas son agregadas al Object Manager y replicadas a los otros clientes que requieran dichos objetos. De esta manera, los usuarios no necesitan tener constantemente instalados localmente todos los componentes disponibles, sólo necesita descargar un conjunto limitado de componentes. Esta extensibilidad dinámica también es de gran importancia cuando se quiere dar soporte a dispositivos móviles o portátiles, que frecuentemente tienen capacidad limitada de almacenamiento.

6.2.5 INTEGRACIÓN WEB

Un requerimiento importante de las herramientas de colaboración modernas es que puedan ser usadas en Internet/Intranet. Este requerimiento va más allá de usar simplemente TCP/IP como protocolo de entrega. Más bien, debe ser posible integrar la funcionalidad colaborativa en las páginas Web, para permitir la cooperación usando los browsers normales de Internet que muchos usuarios tienen disponibles hoy en día.

El cliente DyCE está implementado en Java puro; por lo tanto, puede usarse automáticamente en los browsers de Internet que proveen un ambiente de ejecución Java. Una restricción a esto es la política sandbox de Java, la cual restringe a qué servidores se pueden conectar los Applets Java descargados (específicamente, los Applets Java sólo pueden abrir conexiones de red en el servidor desde el cual fueron descargados inicialmente).

Esta es una restricción seria para los sistemas cliente/servidor distribuidos. Como vimos anteriormente, el servidor DyCE incluye su propio servidor HTTP. Este servidor puede ser usado para prestar servicio a las paginas Web, que contienen el Escritorio Groupware de DyCE o cualquier otra extensión específica del cliente como puede ser un Applet. De esta manera, DyCE puede usarse para “distribuirse a sí mismo” a los usuarios de browsers de Internet. Entonces el Escritorio Groupware, dado que se descarga desde el servidor DyCE, ahora es capaz de abrir conexiones de red en el servidor DyCE y permitir a los usuarios participar en la colaboración dentro de sus browsers Web.

Esta tecnología también puede usarse en componentes DyCE embebidos como pueden ser los Applets dentro páginas Web, realzando así las páginas Web; por ejemplo con componentes de awareness sincrónicos, que pueden ser usados para iniciar la colaboración con otros usuarios DyCE a través de las páginas Web.

6.3 MODELO DE PROGRAMACIÓN DE COMPONENTES

Un objetivo de diseño del sistema DyCE es proveer soporte de manera flexible a la reutilización y la combinación de componentes, así como también a la extensibilidad del sistema en tiempo de ejecución.

Para permitir la reusabilidad y la combinación de Componentes Groupware, se requiere que exista en el framework un modelo de programación general que provea la manera para que los componentes interactúen (por ejemplo, invocarse uno al otro) sin tener, necesariamente, conocimiento mutuo previamente.

Esta extensibilidad es un requisito importante de muchos frameworks de aplicaciones, tal como el framework OLE-2, que es la base de las aplicaciones de oficina comunes. En el framework JavaBeans, la extensibilidad está dada proporcionando acceso a implementaciones de clases descargadas dinámicamente, que pueden ser consultadas a través de un proceso llamado introspección, e invocadas en tiempo de ejecución.

Aún con este soporte, resta el tema de identificar los Beans apropiadamente para la tarea que el usuario quiere ejecutar actualmente. Se requiere un medio para identificar el componente que es requerido por cada usuario en su situación de

colaboración. Para soportar esto, los componentes desarrollados sobre el framework DyCE deben adherirse a un modelo de programación común.

El concepto central del modelo de programación común es la *tarea*, de aquí que el modelo de programación de DyCE sea llamado “*basado en tarea*”. Una tarea es una interacción específica con un ítem de datos para el cual hay disponible en el sistema un Componente Groupware. Las tareas tienen un identificador, un nombre, y se aplican a ciertas clases de elementos del modelo de dominio. Proveen la conexión entre los Componentes Groupware y los objetos de dominio utilizados por los componentes (Figura 4.3).

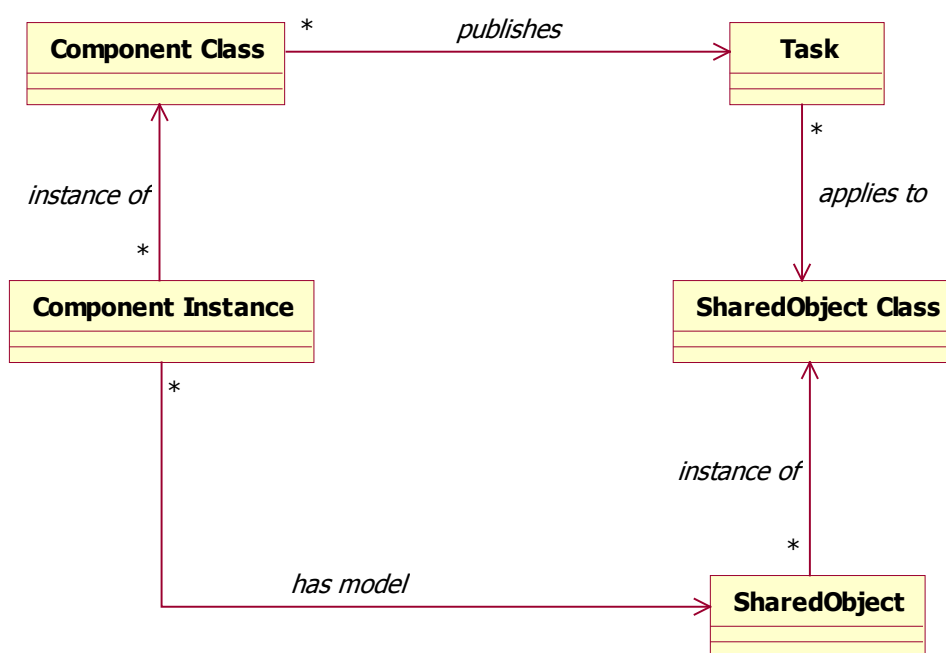


Figura 6.3 Modelo de Tarea DyCE

Cada Componente Groupware *publica* un conjunto de tareas, que pueden ser invocadas por otros componentes o directamente por el usuario (a través del uso de ciertas características en el Escritorio Groupware).

Cuando se usa información de la tarea para acceder a un componente compartido, se dice que la tarea es *ejecutada*. El conjunto de tareas publicadas es manejado por el Component Broker del servidor. De aquí, que el conjunto de tareas también pueden ser consultado para encontrar el componente pertinente para una tarea específica. El ejecutar tareas e invocar a otros componentes puede ser parte de la implementación de un componente. Cuando un componente necesite del servicio de otro componente, lo invocará usando cierta tarea sobre cierto objeto de dato compartido. El framework DyCE provee, a los desarrolladores de componentes, las interfaces apropiadas par hacer uso de esta funcionalidad. La otra manera de invocar una tarea sobre un objeto compartido e por medio del usuario: el Escritorio Groupware provee acceso a todos los

objetos compartidos y permite que cualquier objeto compartido sea depositado como un icono en el escritorio del usuario.

El Component Broker puede utilizar, en tiempo de ejecución, la información sobre la tarea a ser ejecutada, el usuario y la infraestructura para determinar los componentes correctos en cada situación. De esta manera, diferentes usuarios podrían recibir diferentes componentes para la misma tarea (por ejemplo, porque tienen diferentes roles o permisos de acceso).

6.3.1 APLICACIONES Y SESIONES

En DyCE, las sesiones mantienen información sobre el conjunto de usuarios involucrados en la sesión y el conjunto de tareas ejecutadas por ellos. Con el fin de permitir la configuración de ambientes de aplicaciones complejas, se pueden ejecutar múltiples tareas dentro de una sesión simple.

Todos los usuarios de una sesión comparten el mismo conjunto de objetos del modelo y las mismas tareas (pero no necesita estar usando exactamente los mismos componentes para estas tareas). Cuando un usuario o un componente ejecuta una tarea sobre un objeto del modelo dentro de una sesión, el Component Broker del servidor retorna el componente relacionado a esta tarea, que es descargado en el cliente e instanciado localmente en el objeto del modelo. Esta ejecución de las tareas es realizada por todos los usuarios de una sesión, así que cada usuario tiene acceso a los componentes relacionados con la tarea ejecutada. Dependiendo de los roles, derechos de acceso o información de la infraestructura, la ejecución de la misma tarea puede producir diferentes componentes para diferentes usuarios. Sin embargo, los usuarios comparten la misma sesión y objetos del modelo.

Cuando un rezagado se une a una sesión activa, se replica la información de la sesión en la máquina del cliente y se invocan todas las tareas incluidas en la misma.

Debido a que las sesiones de DyCE son modeladas como objetos compartidos y, por consiguiente, son replicadas dinámicamente y persistentes, las sesiones de colaboración pueden ser fácilmente almacenadas, suspendidas y posteriormente reiniciadas.

Cuando se reinicia una sesión suspendida, se ejecutan nuevamente todas las tareas incluidas en la sesión. Si mientras tanto hubo nuevas versiones de componentes, éstos serán descargados e instanciados dinámicamente. De esta manera, los usuarios no están sobrecargados con la necesidad de actualizar constantemente sus ambientes, y además está facilitada la publicación de los nuevos componentes.

6.3.2 SESIONES Y ACOPLAMIENTO

El acoplamiento de los componentes que son usados en una sesión está definido por los componentes. El framework DyCE asegura que los objetos compartidos replicados permanecen en un estado consistente globalmente en el sistema distribuido. El hecho que los usuarios pueden estar usando diferentes componentes para sus tareas, quita todas las posibilidades de utilizar un acoplamiento WYSIWIS estricto. Si ciertas características de la pantalla no están acopladas, necesitan estar desarrolladas como parte del modelo compartido.

6.4 ¿POR QUÉ DYCE?

Lo expuesto hasta aquí sobre DyCE son motivos suficientes para elegirlo como base para el desarrollo de nuestro trabajo. De todos modos haremos un breve resumen de sus características principales.

DyCE es un framework para el desarrollo de groupware basado en componentes, nos permite desarrollar nuevos componentes colaborativos, llamados Componentes Groupware, los cuales son usados en ambientes colaborativos flexibles.

Provee soporte para la extensibilidad del sistema y una fácil publicación de los componentes recientemente desarrollados. Los componentes están basados en un modelo de programación común, el modelo de programación basado en tareas. Este modelo permite la identificación de los componentes relevantes para una tarea y soporta la combinación e integración de componentes, los cuales no tienen conocimiento previo unos de otros. El modelo de programación basado en tareas también sirve como base para la adaptación interactiva colaborativa del sistema por parte del usuario final, por medio del uso de una herramienta de configuración de componentes.

Volviendo a los requerimientos planteados inicialmente, DyCE los lleva a la práctica de la siguiente manera.

La combinación y el rehúso de las herramientas se ven simplificadas debido a que los componentes están basados en un modelo de programación basado en tareas. La estricta separación de los objetos del modelo de dominio y las implementaciones de los componentes permiten el uso de diferentes componentes en el mismo modelo de objeto compartido.

La extensibilidad del sistema en tiempo de ejecución es posible a través del Component Repository, en el cual pueden cargarse los nuevos componentes, y luego, ser descargados e instanciados en el Groupware Desktop. La adaptación del sistema por parte de los usuarios finales, se ve simplificada por una herramienta de composición gráfica, donde los usuarios pueden crear colaborativamente configuraciones compartidas, incluyendo componentes múltiples.

La independencia de plataforma es soportada debido a que DyCE está realizado en el lenguaje de programación Java. Los clientes DyCE pueden ser enviados a través de Internet/Intranet, embebidos en páginas Web servidas por el servidor de DyCE.

Provee soporte para diversos modos de colaboración. La colaboración sincrónica y asincrónica es soportada debido a que todos los objetos compartidos son persistentes automáticamente y pueden ser recuperados desde el almacenamiento persistente cuando sean necesarios. El soporte para el trabajo mono-usuario, grupal, como así también, para las transiciones entre éstos, es provisto por el modelo de sesiones, donde un usuario puede crear un nuevo ítem, comenzar a trabajar en él usando componentes DyCE, y más tarde invitar a otros usuarios a su sesión de edición, ocasionando, automáticamente, los objetos compartidos y los componentes relevantes disponibles a ellos.

6.5 CHATBLOCKS

Chatblocks es un framework de caja blanca orientado a objetos desarrollado sobre DyCE, por este motivo una aplicación final creada con Chatblocks conforma un Componente Groupware en DyCE.

El dominio específico de este framework está orientado a proporcionar un soporte para la creación de herramientas de comunicación sincrónica basadas especialmente en texto, conocidas dentro de los ambientes groupware como Sistemas “Chat”.

Como mencionamos anteriormente, nuestro trabajo está desarrollado sobre DyCE, y por lo tanto también conforma un Componente Groupware en DyCE. Además, los componentes groupware pueden acoplarse para formar aplicaciones más complejas e interconectadas.

Dentro de nuestro trabajo existe un bloque de debate (del cual hablaremos más en detalle en los capítulos siguientes), en el cual es necesario que los participantes discutan y negocien. Para esto es necesario que los usuarios intercambien mensajes, lo cual puede verse inicialmente como un “sistema de chat”.

Elegimos utilizar Chatblocks para desarrollar el debate, ya que nos proporciona la funcionalidad necesaria para implementar un sistema de chat de acuerdo a nuestras

necesidades, y además, puede integrarse con el resto del trabajo ya que todo está desarrollado sobre el mismo framework, DyCE.

CAPITULO 7 - CONCLUSIONES Y TRABAJO FUTURO

7.1 EXPERIENCIA DE USO

A continuación presentaremos los puntos más relevantes que surgieron a lo largo del desarrollo de nuestro trabajo.

SISTEMA OPERATIVO.

Comenzamos el desarrollo del IWA sobre Windows 98 y Millenium planteando un modelo de datos genérico, fácilmente extensible a otros Sistemas Operativos.

A raíz de los avances en el mercado, ya avanzado el desarrollo de la tesis, nos pareció conveniente que el IWA pudiera ser instalado y ejecutado sobre plataformas más actuales. Con lo cual, decidimos la migración hacia Windows XP, encontrándonos con la sorpresa que todo lo desarrollado hasta el momento funcionaba correctamente, sin necesidad de modificar nada, salvo cambiar las DLLs por las correspondientes al nuevo Sistema Operativo.

JAVA

Por los mismos motivos que migramos de Sistema Operativo, decidimos pasar de la versión 1.3 de Java a la 1.4, sin la necesidad de realizar ningún cambio en el código generado hasta ese momento.

INTERACCIÓN CON EL SISTEMA OPERATIVO

Al investigar como extender la funcionalidad provista por Windows para el manejo del sistema de archivos, analizamos distintas alternativas antes de llegar a la definitiva. Nuestra meta era encontrar algún mecanismo que nos permitiera invocar a las DLLs propias de Windows, ya que las mismas proporcionan la funcionalidad buscada. De esta forma, nos abstraíamos de conocer el manejo interno de la plataforma y evitábamos tener que escribir dichas funciones con un lenguaje de bajo nivel, como puede ser

C/C++. Elegimos este camino dado que considerábamos que realizar dichas implementaciones nos desviaba del objetivo de nuestro trabajo, sin realizar ningún aporte al mismo.

Durante el transcurso de esta investigación encontramos el paquete de clases WFC de Microsoft Visual J++, que provee el acceso a algunas de las DLLs del sistema operativo, pero las mismas no cubrían toda la funcionalidad que necesitábamos para el desarrollo de nuestra aplicación.

Finalmente, encontramos el paquete SFC (Servertec Foundation Classes), que como mencionamos en el capítulo 6, permite el acceso a las funcionalidades nativas. Si bien este paquete se adaptaba a nuestros requerimientos, presentaba el inconveniente que no funcionaba cuando el nombre del archivo involucrado contenía acento.

A partir de esta falencia comenzamos una comunicación con Servertec, en la cual le presentamos nuestra inquietud, y al cabo de unos meses nos notificaron que la nueva versión disponible del paquete resolvía el problema planteado.

7.2 TRABAJO FUTURO

En esta sección mencionaremos algunas de las distintas e interesantes alternativas de desarrollos que pueden realizarse como extensiones de nuestro trabajo.

Cabe aclarar que algunas de estas alternativas podrían haberse incluido en esta tesis, pero no lo creímos conveniente ya que las mismas no contribuían a la idea principal de nuestro trabajo y nos desviaban del objetivo del mismo.

ACTUALIZACIÓN BIDIRECCIONAL ENTRE EL IWA Y EL SISTEMA OPERATIVO

De la misma manera que se actualiza el ambiente de trabajo del sistema operativo ante algún cambio en el ambiente de trabajo del IWA, sería interesante que el IWA refleje las novedades que el usuario realiza directamente sobre el sistema operativo.

Para esto, se debería implementar algún mecanismo que le permita al IWA escuchar la incorporación o eliminación, de herramientas y documentos, realizadas mediante los métodos provistos por la plataforma.

Esta funcionalidad podría implementarse utilizando objetos listeners, destinados a detectar cualquiera de estos cambios. Y así, poder iniciar un debate, al igual que cuando se produce un cambio desde el entorno de trabajo del IWA.

Con esta extensión, se podría llegar a una configuración de trabajo “ideal” ya sea desde el IWA o desde el sistema operativo subyacente.

EXTENSIÓN A OTRAS PLATAFORMAS Y APLICACIONES

Dejamos la puerta abierta para extender la idea planteada, no sólo al entorno de trabajo de otras plataformas, sino también al de otras aplicaciones, como puede ser el portal de una entidad en su Intranet o en Internet. Esto se lograría aplicando en dichas aplicaciones los conceptos de roles, identidades, herramientas, recursos, negociación (discusión), adaptación y evolución.

7.3 ESTADO DEL ARTE

Uno de los problemas que se observan al querer aplicar adaptación en los sistemas de groupware, son los conflictos que surgen de los intereses divergentes del usuario que intenta adaptar alguna función del sistema, y el usuario afectado por la elección del mismo.

La solución a este problema, propuesta por Volker Wulf, es extender los sistemas de groupware con un mecanismo de negociación para cada función del sistema de groupware adaptable.

Por otro lado, para que un sistema de groupware sea considerado exitoso, debe ser usado y aceptado por muchos, sino por todos, los miembros del grupo. Las diferencias presentadas entre los miembros del grupo (la variedad de roles, necesidades, habilidades, etc) y las diferencias entre los distintos grupos, viendo a cada grupo como un todo, son un serio obstáculo para alcanzar la aceptación uniforme del producto de groupware, especialmente, cuando el producto trata a todas las personas y grupos idénticamente. Greenberg plantea alguna de las consecuencias de no tener en cuenta las diferencias individuales:

- La inhabilidad del grupo para alcanzar la masa crítica.
- La accesibilidad desigual de los participantes al sistema de groupware.
- La carencia de definir los diferentes roles y participantes que pueden participar.
- La carencia para balancear el trabajo hecho contra los beneficios recibidos.
- La carencia del groupware de evolucionar con las necesidades del grupo.

Como solución propone la noción de groupware personalizable, definido como un sistema cuyo comportamiento puede ser alterado para adaptarlo a las necesidades particulares de los participantes del grupo y a cada grupo como un todo.

En la actualidad, en los entornos X-Windows los usuarios adaptan su entorno de trabajo (escritorio, menú de acceso rápido, etc.) de acuerdo a las tareas que realizan comúnmente. Dichos entornos de trabajo están basados en identidades, por lo que están determinados por una configuración básica a la que se le agrega la configuración

dependiente de la identidad del usuario.

Si bien, los entornos X-Windows van evolucionando, a medida que el usuario ataca nuevas tareas o desempeña distintos roles, no son aplicables para trabajo en grupo, ya que los mismos no pueden compartirse entre los usuarios, ni evolucionar colaborativamente.

Con la solución planteada en este trabajo, extendemos esta funcionalidad para cubrir los aspectos de colaboración y comunicación necesarios para poder obtener un entorno de trabajo compartido por los miembros de un grupo que desarrollen la misma actividad, y que pueda evolucionar colaborativamente.

Para lograr esto incluimos el concepto de roles, y la posibilidad de adaptar los entornos de trabajo a través de un mecanismo de negociación.

7.4 CONCLUSIONES

Con la implementación del IWA logramos plasmar los conceptos involucrados en nuestra investigación, obteniendo una aplicación desarrollada con tecnología de punta que ofrece los siguientes beneficios a las personas que trabajan dentro de un grupo de trabajo:

- Los usuarios que comparten un mismo rol dentro del grupo evolucionan colaborativamente a una Configuración de Trabajo “Ideal” para dicho rol.
- Con la Negociación/Discusión se obtiene una comunicación que explota los aspectos de la colaboración.
- Se trabaja de una manera más integrada.
- Se intercambia conocimiento y experiencias personales.
- Las personas aprenden unas de otras.
- Se optimizan los tiempos de aprendizajes de los usuarios más novatos.

Con nuestro trabajo pretendemos abrir nuevos focos de investigación en áreas novedosas, como lo son el trabajo colaborativo y el aprendizaje continuo.

Por otro lado, en el mercado actual, existe una movida para construir nuevos entornos de trabajo. Un ejemplo de esto es la herramienta de IBM, WebSphere Portal, que dota al colaborador de un único punto de entrada desde dónde los usuarios pueden comunicarse, colaborar entre si y encontrar el contenido de los servicios que precisan para garantizar el éxito de sus tareas y de su negocio.

ANEXO A - TRABAJO COLABORATIVO Y GROUPWARE

A.1 TAXONOMÍAS DE SISTEMAS DE GROUPWARE

En esta sección mostraremos tres taxonomías útiles para ver la variedad de groupware. La primera esta basada en las nociones de tiempo y espacio, y la segunda en el nivel de funcionalidad de la aplicación.

A.1.1 TAXONOMÍA DE TIEMPO Y ESPACIO

Podemos pensar en groupware para asistir a grupos cara-a-cara o distribuidos físicamente. Además, un sistema de groupware puede ser ideado para mejorar la comunicación y colaboración en interacciones de tiempo real o asincrónicas. Estas consideraciones de tiempo y espacio dan lugar a las cuatro categorías de groupware representados en la matriz de la Figura 3.1.

De esta manera, una sala de reuniones estará ubicada en la celda superior izquierda; un editor de documentos en tiempo real se ubicará en la celda inferior izquierda; una pizarra de novedades, en la celda superior derecha; y un sistema de correo electrónico se ubicará en la celda inferior derecha.

		T I E M P O	
		Sincrónicas	Asincrónicas
E S P A C I O	Locales	Interacción cara-a-cara Ej: Sala de Reuniones	Interacción Asincrónica Ej: pizarra de novedades
	Distribuidas	Interacción distribuida Sincrónica Ej: Editor en Tiempo Real	Interacción distribuida Asincrónica Ej: Correo Electrónico

Figura A.1. Matriz de Tiempo Espacio de Groupware

Un sistema de groupware integral puede servir mejor a las necesidades de todos los cuadrantes. Por ejemplo, puede ser beneficioso tener la misma funcionalidad base y look and feel de la interfase de usuario (a) mientras se está editando un documento en tiempo real con un grupo de personas (mismo tiempo / mismo lugar, o mismo tiempo / diferente lugar) y (b) cuando se está trabajando sólo en la oficina o en la casa (diferentes tiempos).

A.1.2 TAXONOMÍA DE NIVEL DE APLICACIÓN

La segunda taxonomía presentada está basada en el nivel de funcionalidad de las aplicaciones y no significa que sea completa; además, muchas de las categorías definidas se superponen.

Esta taxonomía, ante todo, intenta dar una idea general de la amplitud del dominio de groupware.

Sistemas de Mensajes

Proveen intercambio asincrónico de mensajes textuales entre grupos de usuarios. Los ejemplos más comunes son el correo electrónico o conferencias por computador. La proliferación de estos sistemas ha llevado al fenómeno de “sobrecarga de información”. Algunos sistemas ayudan al manejo de esta información aliviando el proceso de carga del usuario. La “inteligencia” es agregada al sistema de entrega de mensajes o a los mensajes en sí.

Editores Multiusuarios

Los miembros de un grupo pueden usar editores para crear y editar documentos de manera conjunta. Estos editores pueden ser:

Asincrónicos: cada usuario es responsable de una parte del documento, en la cual trabaja dependiendo de sus tiempos.

Sincrónicos: todos los usuarios editan el mismo objeto al mismo tiempo. El objeto que se está editando, generalmente, es dividido en segmentos; por ejemplo un documento puede dividirse en secciones y un programa en procedimientos o módulos. Comúnmente, un editor multiusuario permite acceso de lectura concurrente a cualquier segmento, pero sólo un escritor a la vez. El editor maneja transparentemente el bloqueo y la sincronización, y los usuarios editan objetos compartidos como si fueran objetos privados.

Algunos editores multiusuarios proveen notificación explícita de las acciones de los otros usuarios.

Sistemas de Soporte de Decisiones

Proveen facilidades basadas en computadora para la investigación de problemas no estructurados en un grupo. El objetivo es mejorar la productividad de la toma de decisiones, ya sea, acelerando el proceso de la toma de decisiones o mejorando la

calidad de las decisiones tomadas.

Muchos de estos sistemas se implementan como salas de reuniones electrónicas que contienen varias workstation distribuidas en red, grandes pantallas públicas controladas por computadora y equipamiento de audio y video. Algunas de estas facilidades requieren de un operador especialmente capacitado; otros asumen una competencia operacional entre los integrantes del grupo.

Conferencias por computador

La computación sirve como medio de comunicación en una variedad de formas. En particular, provee tres maneras de llevar a cabo conferencias entre las personas:

Conferencias por computador en tiempo real: permiten a grupos de usuarios, reunidos en una sala de reuniones electrónica o distribuidos físicamente, interactuar sincrónicamente a través de sus workstations.

Existen dos formas de implementar este tipo de conferencias. La primera consiste en embeber una aplicación mono-usuario en un ambiente de conferencia que multiplexa la salida de la aplicación al resto de los participantes. La segunda alternativa consiste en diseñar aplicaciones especialmente para múltiples usuarios.

Las dos opciones presentan ventajas y desventajas. Mientras que la primera implementación permite usar las aplicaciones existentes, cada usuario tiene una vista idéntica de la aplicación, no hay contexto por usuario. La segunda opción ofrece la posibilidad de desarrollar una interfase más rica, pero la aplicación deberá construirse desde cero o con un esfuerzo adicional considerable. Pero ninguna de las dos opciones brinda las facilidades del video.

Teleconferencias: soportan interacciones de grupo a distancia. Los ejemplos más comunes son las videoconferencias.

Las teleconferencias tienden a ser complicadas ya que requieren de salas especiales y, a veces, de operadores especializados. Además, no permiten compartir texto ni gráficos entre los participantes.

Conferencias de escritorio: este tipo de conferencias combina las ventajas de las dos anteriores, tratando de disminuir sus desventajas.

Utiliza la workstation como la interfase de conferencia, pero también ejecuta aplicaciones compartidas por los participantes. Algunos sistemas que implementan este tipo de conferencias soportan múltiples ventanas de video por workstation. Esto permite la presentación de vistas dinámicas de información e imágenes de video de los participantes.

Agentes Inteligentes

Permiten la incorporación de agentes no humanos a un grupo de personas. En general, son responsables de un conjunto de tareas específicas, y la interfase de usuario es la que refleja sus acciones al resto de los participantes.

Sistemas de Coordinación

El problema de la coordinación es “la integración y adecuación armoniosa de los esfuerzos de trabajo individuales para lograr un objetivo aún más grande”. Los sistemas de coordinación abordan este problema de varias maneras.

Generalmente, estos sistemas permiten a los individuos ver sus propias acciones y aquellas relevantes del resto del grupo, dentro del contexto de un objetivo general.

También pueden disparar acciones informando a los usuarios del estado del sistema o generar alertas o recordatorios automáticamente.

Los sistemas de coordinación pueden ser categorizados en cuatro tipos de modelos:

- *Orientado a formularios*: están basados en el ruteo de documentos en procedimientos organizacionales. Estos sistemas dirigen la coordinación modelando explícitamente las actividades de la organización como procesos fijos. En algunos de estos sistemas existe un esfuerzo para hacer que el soporte de los procesos sea más flexible.
- *Orientado a procedimientos*: estos modelos ven a los procesos organizacionales como procesos programables. El desarrollo de estos procesos programables es una tarea rigurosa que está compuesta por las fases de especificación, diseño, implementación y testing o verificación.
- *Orientado a conversación*: en estos modelos las personas coordinan sus actividades vía la conversación
- *Orientado a estructura de comunicación*: estos modelos describen las actividades organizacionales en términos de relaciones entre roles.

Como mencionamos anteriormente, las categorías pueden superponerse. A medida que aumenta la demanda de sistemas integrados, vemos más combinaciones entre las distintas funcionalidades. Por ejemplo, los sistemas de mensajes inteligentes pueden y han sido usados para coordinación. Los sistemas de conferencia de escritorio pueden y han sido usados por grupos de edición. De todos modos, muchos sistemas pueden ser categorizados de acuerdo a su objetivo principal; y estos, a su vez, pueden depender de las perspectivas de los diseñadores de sistemas.

A.1.3 TAXONOMÍA DE NIVEL DE COMPARTICIÓN DE OBJETOS

También se puede hablar de una categorización basada en el nivel de compartición de los objetos de colaboración.

Sistemas basados en Documentos y Formularios

Cuando la colaboración y comunicación implican documentos, archivos de aplicación y

formularios, las principales de una amplia gama de aplicaciones de groupware, caen en esta categoría, tales como el correo electrónico, flujo de trabajo y documentos administrativos.

Sistemas basados en un Alto Volumen Transaccional de Información

Cuando la comunicación y colaboración implican un alto volumen de recuperación de registros o procesos y transaccionales. La mayoría de las aplicaciones colaborativas relacionadas son típicamente construidas para un alto rendimiento administrativo de base de datos, recuperación de información y sistemas de imágenes de documentos. En este caso, los formularios son casi siempre usados como sistemas administradores de base de datos para imágenes de documentos. Además, la mayoría de los sistemas de imágenes de documentos, también incluyen sistemas de flujo de trabajo. En realidad muchos flujos de trabajo y sistemas de administración de documentos usan sistemas administradores de base de datos para almacenar y recuperar objetos en sus respectivas aplicaciones.

La principal diferencia entre estas dos categorías, es ante todo, el volumen, la robustez de información y el tipo de aplicación. En esta riqueza de simplificación, las aplicaciones basadas en un alto volumen transaccional son usados en sistemas basados en productos corporativos donde las transacciones son bien organizadas y definidas para la emisión crítica de datos, el método esta mas o menos definido en un flujo de trabajo.

Los sistemas de comunicación organizacional, se dan cuando las aplicaciones colaborativas aumentan la comunicación y colaboración organizacional. Productos semejantes a calendarios de grupos, video conferencias, reuniones electrónicas y autoría en grupo aumentan el conjunto de comunicación electrónica en una organización. El correo electrónico y el flujo de trabajo entran también en esta categoría, pues son un grupo de productos emergentes que proveen soluciones para la captura de memoria de una organización.

A.2 PERSPECTIVAS

Existen al menos cinco perspectivas o disciplinas que influyen en el desarrollo de sistemas colaborativos: Sistemas Distribuidos, Comunicación, Interacción Humano-Computadora, Inteligencia Artificial y Teoría Social.

La relación entre groupware y estas disciplinas es mutuamente beneficiosa. No sólo, cada disciplina ayuda al avance del entendimiento de la teoría y práctica de groupware; sino que groupware presenta temas de investigación desafiantes en las cinco disciplinas, dichos temas no podrían ser examinados sin groupware.

Generalmente, un sistema de groupware combina las perspectivas de dos o más de estas disciplinas. Por ejemplo, el paradigma de conferencia de escritorio puede obtenerse de cualquiera de las siguientes maneras:

Comenzando con la tecnología de comunicación y mejorándola con computación adicional y pantallas en los teléfonos receptores, o

Comenzando con la workstation personal (perspectiva de sistemas distribuidos) e integrándole las capacidades de las comunicaciones.

Sistemas Distribuidos

Los usuarios están distribuidos en tiempo y/o espacio. Esta perspectiva explora y enfatiza esta descentralización de datos y control. Esencialmente, este tipo de sistemas infiere las propiedades globales del sistema y mantiene la consistencia del estado global observando y manipulando parámetros locales.

Algunos resultados de investigaciones sobre algoritmos eficientes para sistemas operativos y base de datos distribuidos son aplicables a los sistemas de groupware. Por ejemplo, la implementación de sistemas de correo electrónico invoca a aspectos de sistemas distribuidos complejos relacionados con la robustez: los recipientes deben estar disponibles para recibir mensajes, aún cuando el servidor de mails no este disponible. Una solución posible es replicar el almacenamiento de los mensajes en múltiples servidores. El descubrimiento e implementación de los algoritmos requeridos es una tarea desafiante.

Comunicación

Enfatiza el intercambio de información entre agentes remotos. Las principales preocupaciones incluyen el aumento de la conectividad y del ancho de banda, y protocolos para el intercambio de diferentes tipos de información, como texto, gráficos, voz y video.

Uno de los desafíos de groupware en la tecnología de las comunicaciones es implementar interacciones distribuidas tan efectivas como las cara-a-cara.

Interacción Humano-Computadora

Enfatiza la importancia de la interfase de usuario en los sistemas de computación.

La interacción humano-computadora, es en sí misma una multidisciplina, que cuenta con las habilidades de los diseñadores industriales y gráficos, de los expertos en gráficos de computadoras (quienes estudian las tecnologías de display, los dispositivos de entrada y las técnicas de interacción), y los cognitivos científicos (que estudian los conocimientos y percepciones humanas).

Muchas de las investigaciones se focalizaron en sistemas mono-usuarios. El desafío de groupware es ampliar esta perspectiva, dirigiendo los aspectos de la interacción humano-computadora dentro del contexto multiusuario o interfases de grupo. Debido a que estas interfases son sensibles a factores como la dinámica del grupo o la estructura organizacional, son muy importantes los roles que juegan los científicos sociales y los usuarios finales en el desarrollo de las interfases de grupo.

Inteligencia Artificial

En un énfasis en las teorías del comportamiento inteligente, esta perspectiva posiciona a las técnicas de desarrollo y tecnologías para inculcarles a las máquinas atributos humanos. El enfoque de la inteligencia artificial (IA) es usualmente heurístico o aumentativo, permitiendo incrementar la información a través de la interacción usuario-máquina en lugar de estar totalmente completa y estructurada desde un principio.

Este enfoque se combina bien con los requerimientos de groupware. Por ejemplo, el groupware diseñado para usar por diferentes grupos debe ser flexible y adaptable a una variedad de comportamientos y tareas del equipo: los investigadores dan a entender que dos equipos diferentes que ejecutan la misma tarea usan las tecnologías de grupo de diversas formas. Similarmente, un mismo equipo que realiza dos tareas separadas utiliza la tecnología diferentemente para cada tarea.

A lo largo del tiempo, la IA puede ser para el groupware una de las contribuciones mas significativas. Esta tecnología puede transformar máquinas desde agentes pasivos, que procesan y presentan información, a agentes activos que enriquecen las interacciones. El desafío es asegurar que las actividades del sistema, mejoran la interacción en una manera procedural y socialmente deseable para los participantes.

Teoría Social

Esta perspectiva enfatiza la teoría social o sociológica en el diseño de sistemas de groupware. Los sistemas diseñados desde esta perspectiva contienen los principios y explicaciones derivados de la investigación sociológica.

Los sistemas en sí mismos podrían entrenar a los usuarios tanto fomentando como enseñando las teorías sobre las cuales el sistema esta basado.

A.3 CARACTERÍSTICAS DE SISTEMAS COLABORATIVOS

Dentro de un esquema común de trabajo en grupo, generalmente se establecen roles entre los miembros, ya sean implícitos o explícitos, para hacer más eficiente y coordinado el logro de los objetivos. Por otra parte, la información resultante, tanto del proceso del trabajo como del producto final, se conoce como memoria grupal. Además se deben considerar otros factores dentro del esquema de trabajo del grupo tales como los protocolos de colaboración y la forma en cómo cada uno de los miembros del grupo aprecian el trabajo de los demás, lo que constituye la percepción.

Memoria Grupal

Es el espacio común dónde los miembros de un grupo almacenan información en forma ordenada referente al desarrollo de la actividad realizada por el mismo. Este espacio es creado con la finalidad de proveer al grupo de un dispositivo efectivo de comunicación. La Memoria Grupal debe contar con dispositivos de comunicación tanto sincrónicos como asincrónicos, dependiendo del contexto de aplicación de groupware implementado. Es pues, el resultado tanto del proceso del trabajo como del producto final construido por el grupo.

Roles

Un rol es un conjunto de privilegios y responsabilidades atribuidas a una persona o a veces a un módulo del sistema; en este último caso, el rol realizado por un software computacional se denomina agente. Los roles pueden ser atribuidos formal o informalmente. Por ejemplo, a una persona a quien le gusta hablar y relacionarse con muchas personas podría informalmente tomar el rol de guardián de la información. El jefe de un grupo, podría oficialmente tomar el rol de director del grupo.

Dentro de un grupo, los individuos pueden tomar diferentes roles. De este modo, considerando las diferentes parejas que podrían trabajar juntas, tales como autor/editor, emisor/receptor, orador oyente y supervisor/subordinado, observamos que algunas personas podrían cambiar de rol, ser autor en este momento y después tal vez contribuir en la colaboración como editor. Un software podría soportar no solamente a varias personas trabajando en una tarea, sino también diferentes roles.

Protocolos de Colaboración

Son las distintas maneras de interactuar de las personas consensuadas por el grupo. Son reglas que permiten a los individuos comunicarse entre sí de tal forma que cada uno pueda enviar y recibir señales comprensibles para los demás.

Un protocolo de comunicación debe lograr la atención del grupo en el aspecto de la comunicación, identificar los distintos componentes de la comunicación entre las personas, proporcionar retroalimentación constante al grupo de que la comunicación se efectúa satisfactoria o insatisfactoriamente, proporcionar una forma aceptable de concluir la comunicación entre las personas.

Los protocolos grupales o de colaboración pueden ser intuitivos o informales, como los sociales o tecnológicos, o formales como tomar un turno, las reglas de Robert, etc.

Estos protocolos tienen su origen en el entorno social y se hacen necesarios siempre que las personas colaboran en una tarea; se transforman en protocolos de acceso a la información cuando éstas lo hacen mediante el computador, pues el lenguaje, las palabras y los gestos se traducen en datos que necesitan ser almacenados, accedidos, comprendidos y transmitidos.

Percepción o Awareness

Ya que la distancia física disminuye dramáticamente entre los miembros de un grupo, se hace necesario crear mecanismos para proveer información sobre la actividad del grupo. Es por esta razón que casi siempre se asocia el concepto de percepción con los sistemas colaborativos en los que no se da la interacción cara-a-cara, sin embargo los sistemas cara-a-cara también pueden proveer percepción.

Se entiende por percepción, toda información que provee una conciencia grupal al individuo que forma parte de un grupo. La maneja de cómo se obtiene esta conciencia es suministrando información, la cual es interpretada por el usuario dependiendo de sus requerimientos.

Como se refleja intuitivamente, percepción es información sobre la información, se podría hablar de dos tendencias de clasificación, una atendiendo al tipo de objeto de información y la otra atendiendo a la manera en que ésta información es capturada.

En el contexto del objeto de la información tenemos:

Percepción de usuarios: provee información sobre los miembros del grupo, por ejemplo puede informar quienes están conectados y lo que éstos hacen.

Percepción de datos: suministra información referente a los cambios efectuados sobre los datos.

Atendiendo al contexto de la forma en cómo la información se obtiene, tenemos:

Percepción implícita: mediante cámaras de video, altavoces, sensores infrarrojos, teclado, mouse o cualquier método que no requiera la introducción explícita de información en torno a la información por parte del usuario.

Percepción explícita: es provista por los usuarios en forma de calendarios, horarios, letreros como “No molestar” o similares.

La decisión de tomar y proveer un tipo de percepción depende de las situaciones que se deseen implementar y de la estructura de la organización.

A.4 ASPECTOS DE DISEÑO

En el futuro, los sistemas de groupware probablemente incorporarán aspectos de muchas o de todas de las cinco disciplinas vistas anteriormente.

Esta sección pone foco en las investigaciones de groupware, describiendo los problemas que continúan enfrentando los diseñadores y desarrolladores; poniendo énfasis en el groupware de tiempo real diseñado para uso de grupos pequeños y medianos, ya que pensamos, que en estos tipos de grupos, los desafíos técnicos enfrentados por los diseñadores de groupware son más visibles.

A.4.1 INTERFACES GRUPALES

Las interfases grupales introducen problemas de diseño no presentes en las interfases mono-usuario. Un problema básico es como manejar la complejidad: múltiples usuarios pueden producir un nivel más alto de actividad y un mayor grado de concurrencia, y la interfase debe soportar este comportamiento complejo.

Interfaces WYSIWIS (What You See Is What I See)

Tienen como objetivo propagar instancias o aptitudes de los usuarios, garantizan que el contexto compartido se vea igual para todos los participantes.

Las principales ventajas de estas interfases son que tienen un fuerte sentido del contexto compartido y que son de simple implementación; su la mayor desventaja es que pueden ser inflexibles. Son muy difíciles de construir y no facilitan la interacción con el usuario. Existe gran complejidad en el costo del desarrollo, que aumenta mucho más cuando se incrementa la cantidad de usuarios. Se utilizan para sistemas de co-auditoria.

Foco del grupo y aspectos distractivos: una buena interfase de grupo debería describir la actividad del grupo en general y, al mismo tiempo, no ser demasiado distractiva. Esto apunta a la principal diferencia entre las interfases mono y multiusuario. En las interfases monousuario, los usuarios tienen el contexto en mente para interpretar cualquier cambio producido como resultado de sus propias acciones. En cambio, en las interfases multiusuario los usuarios no son conscientes de los contextos de los otros participantes y pueden interpretar con menos facilidad los cambios repentinos producidos como resultados de las acciones de otros.

Aspectos relacionados con la dinámica de grupo: las interfases de grupo deben corresponderse con los patrones de uso del grupo.

Aspectos relacionados con la administración del espacio de la pantalla: el espacio en pantalla es un recurso limitado en aplicaciones monousuario, pero es un problema aún mayor en interfases de grupo, en las cuales cada usuario puede crear ventanas que aparecen sobre las pantallas de otros usuarios. Esto no lleva a la necesidad de técnicas para la administración de la proliferación de ventanas.

Aspectos relacionados con herramientas de interfaces grupales: muchos de los conceptos de interfases simples pueden ser generalizados a interfases multiusuario, por ejemplo, ventanas de grupo y telepointers. Las herramientas de interfases de grupo no deben ser simplemente extensiones de herramientas existentes; mejor dicho, deben introducir nuevos constructores que se acomoden mejor al uso compartido.

A.4.2 PROCESOS DE GRUPOS

Los procesos de grupos son tareas bien definidas que requieren de la participación de un conjunto de usuarios. Frecuentemente, aumentan la sinergia y el paralelismo, pero la sobrecarga de coordinación requerida puede agobiar al grupo y dañar su efectividad. Las tecnologías de groupware pretenden mejorar los beneficios y minimizar la sobrecarga.

Protocolos de grupo

Son acuerdos comunes sobre las maneras de interacción. Los protocolos de grupo pueden ser tecnológicos o sociales.

Protocolos tecnológicos: son contruidos en el hardware y software.

Las principales ventajas de estos protocolos son que aseguran que el proceso sea cumplido, proveen mayor estructura a la actividad del grupo y asisten a los usuarios menos experimentados. Sin embargo, pueden ser excesivamente restrictivos.

Protocolos Sociales: los participantes tienen el control. Incluyen reglas o normas formales y prácticas no tan formales, ambas entendidas y acordadas, pero no forzadas por el sistema.

Estos protocolos fomentan la colaboración, debido a que el grupo debe desarrollar sus propios protocolos, el groupware se hace más adaptable. Sin embargo, pueden ser injustos, distractivos o indeficientes.

Operaciones de grupo

A veces, es apropiado ver el trabajo de múltiples personas como una operación simple. Dichas operaciones reciben el nombre de operaciones de grupo y pueden ocurrir tanto en situaciones sincrónicas como asincrónicas.

Las operaciones de grupo sincrónicas son una de las características que distinguen a groupware de los otros sistemas.

Factores organizacionales y sociales

Es difícil diseñar herramientas que sean útiles en una variedad de situaciones diferentes. Los factores organizacionales y sociales determinan cuanto se ajusta la funcionalidad de la herramienta diseñada a las necesidades de un grupo dado. En general, cuando se especializa una herramienta para satisfacer las necesidades de un grupo en particular, son necesarios conocimientos de grupo y organizacionales.

Excepciones y coordinación

Las excepciones típicas ocurren cuando un usuario no cooperativo falla en completar su rol en la operación, o cuando cambia la composición del grupo.

La coordinación es necesaria ya que las operaciones de grupo imponen obligaciones sobre los participantes.

Una solución simple es dejar que el grupo resuelva tales dificultades usando canales de comunicación alternativos, como puede ser el audio. De todos modos, el sistema debe, al menos, ayudar a detectar problemas y permitir la reconfiguración dinámica de los parámetros de las operaciones, como pueden ser la asignación de roles o el tamaño del grupo.

Integración de apoyo de actividades

Es el soporte de integración de las actividades estructuradas y no estructuradas con las actividades sincrónicas y asincrónicas.

A.4.3 CONTROL DE CONCURRENCIA

Es necesario resolver los conflictos producidos como consecuencia de la participación simultánea de los participantes. Los diseñadores de groupware deben afrontar varios aspectos relacionados con la concurrencia:

Responsiveness

Los sistemas sincrónicos no deben obstaculizar el ritmo del grupo. Para esto se requieren dos propiedades: tiempos de respuesta y de notificación breves. El primer tiempo es el que toma la interfase de cada usuario para reflejar sus propias acciones. Mientras que el segundo, es el tiempo que toman las acciones para ser propagadas al resto de las interfases.

Interfases de grupo

Están basadas en técnicas como WYSIWIS y ventanas de grupo, que requieren que sean mostradas de manera idéntica o casi idéntica.

Si el esquema de control de concurrencia es tal que las acciones producidas por un usuario no son inmediatamente presentadas al resto de los participantes, el efecto de la dinámica del grupo debe ser considerado y sólo aceptar tal esquema si no es destructivo.

Distribución Wide-Área

Uno de los principales beneficios de groupware es que permite a la gente trabajar conjuntamente en tiempo real, aún cuando estén separados por grandes distancias físicas.

Actualmente, con las tecnologías de comunicación existentes, los tiempos y velocidades de transmisión de las redes WAN tienden a ser inferiores que los de las redes LAN. Por este motivo, deben ser considerados los posibles impactos sobre el tiempo de respuesta.

Replicación de datos

Debido a que los sistemas de groupware de tiempo real requieren un corto tiempo de respuesta, sus datos pueden estar replicados localmente. De esta manera, muchas operaciones costosas podrían ser ejecutadas localmente.

Robustness

Se refiere a la recuperación de circunstancias inusuales, tales como fallas de componentes o acciones de usuario impredecibles.

En un ambiente de groupware, también se debe tratar de recuperar las acciones de los usuarios.

A.4.4 OTROS ASPECTOS DE SISTEMAS

Groupware abarca un amplio rango de sistemas. Pero, independientemente del lugar que ocupe el sistema dentro del espectro de groupware, los diseñadores se deben enfrentar a un conjunto común de aspectos de implementación. Algunos de los cuales se presentan a continuación.

Protocolos de comunicación

Una comunicación efectiva es vital para un groupware exitoso. Desgraciadamente, las tecnologías de comunicaciones actuales no son completamente capaces de soportar groupware como uno lo espera.

Primero, las comunicaciones de datos integrados y el audio/video digitalizado no están universalmente disponibles. Los diseñadores de groupware deben utilizar protocolos que se responsabilicen de las diferencias entre los diversos medios.

Un segundo problema es el soporte inadecuado para comunicaciones en grupo. Los protocolos actuales, ya sean circuitos virtuales o basados en datagramas, son más apropiados para comunicaciones entre dos partes.

Control de acceso

Determina quien puede acceder a qué y de que manera. Los permisos no son estáticos, pueden ser modificados o revocados. Un sistema puede simplificar el proceso de obtención de accesos permitiendo la negociación entre las partes.

Los requerimientos de groupware pueden llevar a un modelo de accesos complejo. El acceso a la información cambia frecuentemente, por este motivo, debe haber mecanismos de control de acceso ligeros, que permitan a los usuarios especificar cambios fácilmente.

Notificación

Es la manera de alertar y modificar la interfase de un usuario en respuesta a las acciones ejecutadas por alguien en otra interfase.

En interacciones sincrónicas, la notificación en tiempo real es crítica; de hecho, los tiempos de notificación y de respuesta deben ser comparables.

Existen diferentes granularidades de notificación; en el nivel más fino, cualquier acción de un usuario resulta en una notificación. Mientras que en los niveles más gruesos, varias acciones de usuarios resultan en una notificación.

A.5 HERRAMIENTAS DE GROUPWARE: COMO SE CARACTERIZAN

A.5.1 GROUPKIT

Es un conjunto de herramientas de groupware basadas en el lenguaje Tcl/Tk. Es de uso libre y fue desarrollado por la Universidad de Calgary. Es usado para construir aplicaciones de tiempo real tales como herramientas de dibujo, editores y sistemas de reuniones que pueden ser simultáneamente compartidos por varios usuarios.

Fue originalmente desarrollado para ser usado en ambientes Unix, aunque actualmente existe una versión beta para Windows y Macintosh.

Ha sido usado para crear prototipos de groupware, para investigación de interfaces y arquitecturas multiusuario, y como herramientas de aprendizaje de CSCW.

Algunos ejemplos de aplicaciones que se pueden construir sobre GroupKit son:

Brainstorming Tool

Le permite a los usuarios hacer lluvia de ideas mediante el uso de líneas de texto simples, las cuales aparecen como un listado en la pantalla de todos los usuarios. Las barras de desplazamiento multiusuario ayudan a saber qué parte del documento están viendo los otros usuarios, por ejemplo cuando se discuten ideas.

File Viewer

Una herramienta útil para revisar documentos. Permite cargar un archivo y ver qué parte del documento están mirando los otros usuarios mediante las barras de desplazamiento multiusuario.

Fisheye

Un sistema experimental para tratar de aplicar "fisheye views" al problema de saber dónde están los otros usuarios. Existen múltiples puntos focales, uno por cada usuario. Se puede controlar el despliegue, por ejemplo para mostrar cierta información alrededor de diferentes puntos focales.

Group Sketchpad

Una pizarra compartida que permite dibujar con diferentes colores y anchos de línea. Los telepunteros ayudan a saber donde están trabajando los otros usuarios.

Group Draw

Un programa de dibujo orientado a objetos, donde más que dibujo a mano alzada, objetos como líneas y rectángulos pueden ser creados y modificados luego.

Hello World

La aplicación demo estándar usada para explicar GroupKit. Cuando se presiona el botón, la aplicación dice (por ejemplo) "David dice hola!" a todos los usuarios conectados en la sesión.

Hyper Node Editor

Un editor gráfico que permite crear y manipular nodos. Arcos direccionales pueden ser agregados a cualquier nodo, los cuales siguen al nodo cuando éste es movido.

Postit Note Editor

Un sistema que permite crear y enviar notas postit a uno o más usuarios en una sesión.

Simple Sketchpad

Una versión simple de pizarra de compartida que permite trazos simples, sin otras facilidades como múltiples colores.

Text Editor with Turntaking

Un editor de texto multiusuario que permite asignar turnos de escritura, es decir, sólo un usuario puede hacer cambios en un determinado momento.

Text Chat

Un programa para "chat" similar al "talk" de Unix, que permite escribirles a los otros usuarios en tiempo real.

Text Editor

Un editor de texto experimental que permite a múltiples usuarios escribir al mismo tiempo, mostrando las selecciones (texto seleccionado) de todos los otros usuarios.

A.5.2 LOTUS NOTES

Lotus Notes es una aplicación Cliente-Servidor con facilidades de comunicar y para

compartir información entre grupos de trabajo.

Rol de Lotus Notes en Groupware

- Líder en el Mercado.
- Una plataforma para construir aplicaciones que apunten a información o a procesos (Workflow).
- Aplicaciones de Usuario Final.
- Sistema global de E-mail.
- Sistemas de Conferencias Electrónicas.
- Bases de Datos Distribuidas Semiestructuradas.

Casos típicos de aplicaciones Lotus Notes

Lotus Notes es una herramienta que provee el soporte electrónico para que grupos de personas puedan trabajar en forma individual, hacia un objetivo común, por lo tanto las aplicaciones más comunes son:

- *Comunicación: Propiedad (Authoring), direccionamiento, transmisión y recepción de mensajes y documentos electrónicos. Ejemplos: Electronic Mail, Compound Documents (with attachments), Intercompany comms (World-Com), Internet Access.*
- *Coordinación: Soporte electrónico a grupos que trabajan hacia un objetivo común. Ejemplos: Strategic Direction, Project Management, Financial Discussions, Building Maintenance, Product Evaluations, Interest Groups.*
- *Referencia: Repositorios de información empresarial relevante, que puede ser accedida electrónicamente por múltiples usuarios para ayudar en la toma de sus decisiones. Ejemplos: Corporate Procedures, Project Standards, Technical Guidelines, Contract Database, Manuales, Directorios.*
- *Colaboración/Workflow: Soporte electrónico para realizar procesos específicos (que pueden requerir varios pasos), que están directamente relacionados con los objetivos de la Empresa. Ejemplos: Purchase Order, Authorizations, Customer Support, Invoice Approval, Travel Approval, Document Control.*
- *Seguimiento (Tracking): Historia y estado del monitor de procesos, eventos, incidentes, planificaciones (schedules). Ejemplos: Employee Records, Project Status, Account History, Competitor Activity, Customer Service, Sales Activity, Help Desk, Meeting Rooms, Application Requests.*
- *Anuncios (Broadcast): Comunicación de la misma información, al mismo tiempo, a grandes cantidades de usuarios. Ejemplos: Industry News, Current Events, Corporate News, Corporate Announcements.*

Desafíos para implementar Sistemas de Groupware

Cambios Organizacionales

- Definición de Objetivos de Negocios.
- Definición de políticas de modernización.
- Trabajo en equipo.
- Soporte sobre la marcha.

Desafíos Culturales: transformación de una Cultura Competitiva e Individualista a Esfuerzo Compartido, Colaboración y Cooperación; utilizando políticas apropiadas.

Desafíos Tecnológicos

- Ambientes heterogéneos.
- Arquitectura de almacenamiento y reenvío.
- Sincronización de directorios.
- Bases de datos replicadas.
- Seguridad.
- Interfaz de usuario intuitiva.
- Buen desempeño (performance) de la B.D.
- Almacenamiento adecuado de los datos.
- Configurable para el Usuario Final.
- Soporte para estructuras.

A.5.3 HABANERO

El proyecto Habanero de NCSA investiga facilidades para convertir aplicaciones monousuario en multiusuario, creando ambientes de trabajo colaborativo.

Es un marco de trabajo para compartir objetos Java con colegas distribuidos a través de Internet. Incluye facilidades de autenticación y privacidad. El software es de libre distribución.

Algunos ejemplos de aplicaciones construidas usando Habanero son:

The Habanero WhiteBoard

El Habanero WhiteBoard puede ser usado para compartir imágenes y dibujos durante una sesión. Se pueden compartir imágenes GIF y JPEG, tanto desde un disco local o desde cualquier punto en el Web. Se cuenta con herramientas de dibujo para hacer o anotar las imágenes.

The Collaborative Calculator

Esta calculadora postfija ofrece las operaciones básicas más algunas funciones más complejas. Presenta un visor para cada usuario.

The H263 Video Player

Esta realizado completamente en Java, usando un formato de archivos diseñado exclusivamente para éste. Es capaz de abrir archivos locales y remotos mediante direcciones URL. El sistema puede manejar hasta 3 frames por segundo sobre un modem de 28.8 corriendo en Windows 95. Muestra el mismo video a todos los usuarios de la sesión.

The Sound Player

Está realizado completamente en Java, y soporta el formato GSM. Este formato ha sido optimizado para voz. Puede leer archivos locales y remotos a través de direcciones URL.

The Voting Tool

Brinda democracia a una sesión colaborativa. El iniciador de la votación define el tema a votar, el tipo de respuesta (Sí/No, %, escogencia múltiple), las opciones, quién recibe los resultados, y si los votos se hace anónimos o no. Después de la votación cada usuario recibe una ventana con el resultado.

The Collaborative Text Editor

Este prototipo permite abrir un archivo local, guardar el archivo actual en un disco local, distinguir cada usuario mediante un color de letra distinto, desplegar el nombre de cada colaborador, definir el tipo de letra (bold, italic o normal).

The AudioChat

Es básicamente un foro basado en audio. Los participantes de una sesión Habanero pueden usar esta herramienta para comunicarse con los otros de manera organizada. Mantiene una lista de usuarios que desean hablar, asegurándose de que sólo un usuario hable a la vez.

Color Drawing

Es una pizarra compartida.

The Visible Human

Permite recuperar y mostrar imágenes médicas de una base de datos. Es una aplicación colaborativa la cual tiene un panel de control que puede ser usado desde cualquier estación de trabajo en la sesión. Mediante estos controles los colaboradores pueden recuperar imágenes de la base de datos. Todos los usuarios ven las mismas imágenes y pueden hacer anotaciones sobre las imágenes. Esta aplicación fue desarrollada para apoyar discusiones entre profesionales del área de medicina. También puede ser usada para apoyar la creación de un salón de clases de medicina virtual.

A.6 EJEMPLOS DE APLICACIONES COLABORATIVAS

Como vimos anteriormente, el objetivo de CSCW es proveer un mejor soporte a la colaboración provista por sistemas tradicionales. El soporte actual para la colaboración consiste básicamente en charlas, correo electrónico, y archivos y bases de datos. Entender qué más se necesita, nos permitió comparar estas aplicaciones, identificando las propiedades que distinguen unas de otras.

Una aplicación de la charla soporta la abstracción de una sesión común en la que los usuarios pueden entrar o salir. El texto ingresado por un usuario no es destinado a un usuario específico, sino que todos los usuarios en la sesión son los receptores implícitos del mismo. Es más, la aplicación hace el mayor esfuerzo para comunicar lo ingresado de por un usuario, tan pronto como sea posible, manteniendo activas las conexiones entre todos los usuarios. Esta aplicación es adecuada para soportar reuniones sincrónicas distribuidas.

En contraste, el correo no tiene noción de la existencia de una sesión y requiere destinatarios de mensaje específicos. Es más, no mantiene activas las conexiones, y encolar y archivar mensajes para disminuir el tráfico de la red. De esta manera, se obtiene una colaboración asíncrona, aunque es posible usarlo (ineficientemente) para una colaboración sincrónica.

Tanto en los sistemas de charla como en los de correo, la información se copia explícitamente en los espacios privados de los colaboradores.

Por otro lado, los sistemas de archivos o base de datos, crean una única copia compartida de un mismo objeto, y la comunicación ocurre indirectamente supervisando el estado de este objeto. Así, un archivo es conveniente para colaboración que involucra la creación de un objeto compartido.

De esta manera, estos tres sistemas definen tres clases de aplicaciones, aplicaciones *basadas en sesiones*, *basadas en mensajes*, y *basadas en objetos*. Podemos mejorar el estado del arte (a) manteniendo un mejor soporte para cada tipo de sistema, conservando las propiedades básicas de cada uno de ellos, y (b) combinando las propiedades de estos sistemas, y así, creando nuevos, híbridos, tipos de aplicaciones colaborativas.

Primero veremos las aplicaciones que proveen soporte a cada uno de estos tipos de sistemas, y luego, aquellas que combinan las propiedades de dichos tipos. Esta clasificación intenta dar una herramienta para organizar el tour de aplicaciones colaborativas. No significa que sea tomada estrictamente, ya que muchas aplicaciones combinan las propiedades de múltiples tipos de sistemas. El tipo en la cual introducimos una aplicación depende de cómo ésta contribuye al flujo de la discusión. Finalmente, ésta es sólo manera de clasificar aplicaciones colaborativas, permitiéndonos derivar nuevas aplicaciones colaborativas basadas en las características de aplicaciones familiares.

A.6.1 COLABORACIÓN BASADA EN SESIONES (TALK++)

MUDs: textual Virtual Reality

La charla es una herramienta pobre para soportar el principio de “estar ahí”, mencionando anteriormente en este capítulo. MUD (Multiuser Dungeons) muestra cómo es posible proveer un soporte más sofisticado para este principio, creando un ambiente virtual que simula un ambiente de reunión físico. A diferencia de otras aplicaciones que veremos más adelante, la intención original no es soportar grupos y objetivos predefinidos, sino proveer una reunión para usuarios que no se conocen entre sí, que podrían participar en juegos, socializar, o hacer algún trabajo real. MUD provee la abstracción de salas y permite a los usuarios ingresar y salir de ellas.

Cuando un usuario ingresa o deja una sala o conferencia, se muestran mensajes, informando lo ocurrido en las pantallas del resto de los usuarios. El nombre y género que identifican a los usuarios son los que ellos informaron explícitamente en MUD, que pueden no ser reales.

MUD es extensible, permite a los usuarios agregar sus propios objetos a la base de datos, tales como salas, salidas y notas. Su idea es combinar la colaboración basada en objetos con el manejo de sesiones.

DIVE: 3-D VR

MUD es un ejemplo de aplicaciones “VR” textuales. DIVE (Distributed Interactive Virtual Environment), en cambio, muestra como se puede proveer una aplicación VR de manera más gráfica para soporte de reuniones; en dicha aplicación, cada usuario o aplicación (agente) se representa por un icono, que puede moverse en un espacio 3D.

El icono tiene asociado un “aura”, que es el espacio alrededor de él. Esta noción de aura es usada para establecer una comunicación entre agentes, la comunicación se establece entre dos partes cuando los iconos de sus auras se superponen. El establecimiento de la comunicación es transitivo: es decir, si el aura del usuario A se intercepta con el aura de B, y el aura de B con el de C, entonces A y C puede comunicarse.

Video Walls

MUD y DIVE crean ambientes artificiales, que tienen características propias de un ambiente de reunión físico. Sin embargo, dicho ambiente es independiente de los ambientes físicos reales de los usuarios en la sesión; de esta manera, el usuario puede no tener la idea de “estar ahí”.

Una solución es usar Video Walls. Estos proveen conexiones de audio y video entre dos salas remotas, en cada sala se ubica un “video wall”, que muestra las actividades llevadas a cabo en la otra sala, y permite la interacción entre las dos partes. El audio se logra colocando parlantes y micrófonos en ambas salas; y el video se logra por medio de cámaras que envían las imágenes a grandes pantallas. Se han realizado pruebas que

demuestran que es posible utilizar esta tecnología para conectar dos lugares geográficos distantes.

Audio and Video Conferencing

La conferencia de audio y video es una extensión del Video Walls. Permite que más de dos usuarios se comuniquen en una misma sesión. Cada usuario ve los videos de sus colaboradores en ventanas separadas y puede hablar con ellos usando distintos canales de audio.

Media Space

Permite a los usuarios “caminar” a través de salas distribuidas; para lograr esto, combina la video conferencia con el paradigma de MUD de navegar a través de las salas.

Las imágenes capturadas por cámaras en los cuartos son intercambiadas por software del servidor, que está ubicado en una de las workstations. Los participantes autorizados pueden pedir al servidor que traiga a sus workstations imágenes de salas remotos, y así tomar conciencia de sus actividades, tal como un usuario es consciente de las actividades de un colaborador cuando la puerta de su oficina está abierta. Esto es útil, por ejemplo, cuando un usuario desea establecer que es un colaborador que puede ser interrumpido.

Portholes: Quasi-Dynamic Media Space

Si comparamos a Media Space con las páginas web que contienen imágenes estáticas, vemos que el primero es más dinámico, ya que permite a los usuarios recibir información actualizada de sus colaboradores.

El concepto de Portholes muestra que existe un punto intermedio entre las imágenes estáticas y el video dinámico: en lugar de enviar imágenes continuamente, pueden enviarse cada tantos minutos. Nynex Portholes extiende esta idea guardando un valor de la hora de las imágenes y permitiendo a los usuarios animarlas. Más aún, permite a los usuarios censurar ciertos períodos de la animación.

Hydra: Gaze Awareness

Los sistemas de video conferencias tradicionales no soportan “gaze awarness”; esto es, no permiten que un usuario tenga contacto visual con un colaborador. Cuando un usuario mira la cámara, no queda claro a qué colaborador dirige la comunicación.

Hydra soluciona este problema facilitando dispositivos especiales, cada uno de los cuales está compuesto por una cámara y una pantalla. Cada colaborador remoto es asignado a un dispositivo que muestra su imagen y, la cámara contenida en el dispositivo captura las imágenes transmitidas a él. De esta manera, cuando un usuario se vuelve hacia un dispositivo, todos los colaboradores saben a quien está mirando.

Un micrófono simple captura el audio local transmitido al resto de los usuarios y el audio de los diferentes colaboradores remotos es combinado en uno solo que es reproducido por un parlante simple. De esta manera, Hydra crea la ilusión de “estar ahí”, aunque

en ciertos casos no se cumple del todo. Los dispositivos de Hydra son pequeños, esto hace que los usuarios vean imágenes diminutas de sus colaboradores. Además, el sistema no crea la ilusión de que los colaboradores estén ubicados en una misma sala y, el micrófono y parlante no permiten el audio direccional.

MAJIC: Seamless User Boundaries

MAJIC extiende el concepto de Hydra y soluciona sus limitaciones dando a los usuarios de una misma sesión la ilusión de estar sentados alrededor en una mesa redonda proyectada en una pantalla especial de forma curva. Cada colaborador es asignado a un arco de la pantalla, en el cual se proyecta su imagen; cada uno de dichos arcos es compuesto por una cámara y un micrófono, que capturan las imágenes y audio transmitidas al colaborador, y un parlante, que reproduce su audio.

GestureCam: Remote Surrogate

En MAJIC e Hydra, las cámaras no pueden ser controladas por usuarios remotos. Por lo tanto, un usuario no puede apuntar o mirar un objeto específico en el cuarto de un colaborador remoto.

Para solucionar este problema, GestureCam ubica un dispositivo en cada colaborador que puede ser controlado por el usuario remoto. El dispositivo tiene tres grados de libertad, posee una cámara y un indicador láser que apuntan en la misma dirección que la cámara. El colaborador remoto puede mover la cámara en tres dimensiones para ver y apuntar a diferentes objetos de la sala. Cuando un usuario toca un objeto mostrado en la pantalla, el dispositivo remoto automáticamente apunta hacia él.

Este sistema se ha usado para asesorar a operadores, por ejemplo, en cómo conectar dos computadoras portátiles, y cómo compartir archivos entre dos computadoras.

Telepresence

Si bien los sistemas se acercan bastante, no soportan del todo la ilusión de estar realmente en un sitio remoto. En particular, un usuario no está inmerso en el ambiente remoto y no puede manipular físicamente los objetos de aquel ambiente. Éste es el objetivo de la telepresence, que usa un “mar de cámaras” para crear dinámicamente un ambiente remoto como un ambiente virtual del sitio local.

3-D Simulation

El espacio compartido no necesita ser artificial o estar basado en los ambientes físicos de los colaboradores, podría ser una simulación de algún objeto físico que necesite ser manipulado por múltiples usuarios. Una opción es proveer a los usuarios una interfase para manipular conjuntamente objetos 3-D; de esta forma, cada usuario ve los cambios realizados por el resto.

Internet Foyer: Mixed Reality

En los ejemplos anteriores vimos dos tipos de sistemas: algunos construyen un espacio artificial en el que los usuarios se encuentran y otros proporcionan un espacio virtual basado en un ambiente físico. Internet Foyer muestra cómo estos dos paradigmas pueden combinarse en una realidad mixta que permite a uno ir “más allá de estar ahí”. El sistema intenta unir el concepto de un vestíbulo físico con uno electrónico. Un vestíbulo puede ser considerado la región pública en algún espacio que informa a los visitantes sobre lo que está disponible en ese espacio y protege el acceso a las regiones privadas. Un vestíbulo físico simplemente es un vestíbulo en un espacio físico (por ejemplo, en un edificio), mientras un vestíbulo virtual es un vestíbulo en algún espacio electrónico.

Por ejemplo, un vestíbulo virtual puede ser un conjunto de páginas web de una organización, que brinda información sobre la empresa y proporciona mecanismos protegidos con contraseñas para acceder a dicha información. La misma organización puede tener un vestíbulo físico en un edificio y uno virtual en web, e *Internet Foyer* muestra cómo pueden combinarse estos conceptos.

A.6.2 COLABORACIÓN BASADA EN MENSAJES (MAIL++)

Information Lens: Typed Messages

A menudo creamos diferentes tipos de mensajes. Sin embargo, el correo tradicional soporta mensajes sin tipo y sin estructura; esto no nos permite capturar las diferencias entre los distintos tipos de mensajes. *Information Lens* resuelve este problema permitiendo a los usuarios definir los tipos de mensajes y organizando dichos tipos en una jerarquía.

En general, el tipado de mensajes ofrece varios beneficios: el tipo define una estructura común a todas sus instancias y puede usarse para filtrar los mensajes a mostrar en una casilla de correo.

CoordinatorWorkflow: Structured Conversations

No sólo podemos asociar una estructura al contenido de un mensaje, sino también a nuestras conversaciones. El *Coordinator* muestra cómo puede capturarse la estructura de una conversación en el sistema del correo electrónico. Es una herramienta basada en un conjunto de disertación de acciones (requerimientos, promesas, etc.) y contiene un modelo de conversación. A medida que los usuarios proponen la construcción conversacional, típicamente mediante el correo electrónico, el sistema genera una trayectoria de sus requerimientos y compromisos. Está basado en el sistema de mensajes, fue diseñado para facilitar el intercambio, clarificación y negociación de compromisos en las organizaciones. Existen pocas dudas de que esta herramienta fue proyectada para facilitar el trabajo cooperativo, como un medio para lograr la democracia y el respeto por la gente en las organizaciones. Mientras algunos reportes señalan una amplia aceptación en las empresas, otros lo critican fuertemente por su

desempeño y naturaleza. Hay además, reportes que mencionan que contrariamente a lo esperado, la herramienta sólo ha sido adoptada en organizaciones de tipo jerárquico.

Computational Mail

El mail computacional es un concepto similar al workflow, ya que asocia operaciones con acciones definidas por el usuario.

En el mail computacional, además de enviar datos, se pueden enviar programas ejecutables, que son activados por el recipiente del mensaje. Dichos programas pueden ser usados para mostrar información al usuario, o para recolectar información del mismo. Por este motivo, el mail computacional debe asegurar portabilidad y seguridad, para que la ejecución de los programas no cause problemas en las máquinas que reciben el mail.

A.6.3 COLABORACIÓN BASADA EN OBJETOS (FILE++)

Estos sistemas permiten a los colaboradores compartir objetos lógicos y proveen soporte especializado para la colaboración.

Hypermedia

El éxito de la Web ha mostrado las diversas maneras en que los hyperlinks pueden ser útiles en la colaboración distribuida. Por ejemplo, facilitan la asociación de un objeto con sus comentarios o análisis de diseño.

Quilt: Writeable Hypermedia and Typed Links

Quilt es un ejemplo de un sistema que soporta hyperlinks tipados y objetos modificables (características que la Web no soporta). Fue diseñado como soporte para permitir la escritura colaborativamente sobre un documento.

Un documento Quilt tiene una estructura de árbol, esta compuesto por un nodo raíz y varios nodos de anotaciones. Las anotaciones pueden ser predefinidas o definidas por el usuario, y pueden ser usadas en consultas realizadas en diferentes vistas del documento.

Soporta diferentes roles, cada uno de los cuales tiene asociado un conjunto de permisos. Cada usuario toma un rol particular y se le asignan los permisos asociados a dicho rol.

También provee casillas de correo para cada usuario, permitiéndoles ver anotaciones dirigidas a ellos sin la necesidad de acceder al documento. Al igual que las bases de datos activas, permite asociar acciones definidas por los usuarios con ciertas condiciones. De esta manera, Quilt también es un ejemplo de un sistema que combina colaboración basada en mensajes y objetos.

PREP: Zero-Cost Hyperlinks

En los sistemas de hypermedia, el costo de pasar por un link es bajo, pero no es cero. El editor y base de datos PREP ilustra como puede eliminarse dicho costo.

La base de datos almacena secciones o bloques de datos y permite a los usuarios agrupar dichas secciones en borradores. Cada borrador corresponde a algún documento compartido en el que los usuarios están colaborando. El editor muestra en forma de matriz las secciones que componen los borradores. Generalmente, una columna contiene parte del documento y las otras contienen los comentarios realizados por los distintos usuarios sobre las columnas a su izquierda. De esta manera, leyendo de izquierda a derecha, se ve la discusión sobre el progreso de una parte del documento; y de arriba hacia abajo, se ven los comentarios realizados por un usuario en particular sobre las diferentes secciones del documento.

PREP puede ser considerada como un caso especial de un sistema de hipertexto, dónde un comentario puede considerarse como un nodo ligado a alguna sección de datos sobre la que se hace el comentario. A diferencia de los sistemas tradicionales de hipertexto, PREP no requiere de esfuerzo por parte del usuario para navegar por los hiperlinks, debido a que los comentarios están siempre visibles.

Las características descritas en esta sección se aplican a la versión inicial de PREP.

IBIS: Structured Discussion

Las columnas de PREP permiten organizar de una única manera la discusión sobre un tema. IBIS (Issue-based Information System) emplea un método más sofisticado para capturar los fundamentos que están detrás de las decisiones. Soporta las nociones de temas, posiciones de dichos temas, argumentos a favor y en contra de las posiciones y captura las relaciones entre ellos. IBIS es una herramienta textual que muestra estas relaciones de una manera indentada.

CLARE: Structured Discussion + Process Model

IBIS permite estructurar las discusiones de una única manera. Por otro lado, CLARE (Collaborative Research And Research Environment) ofrece una mayor estructura para organizar una discusión e integrarla con el modelo de proceso. Está formada por dos componentes:

RESTR (REpresentation Schema of Research Artifacts), define la estructura de una discusión mediante varios tipos de nodos enlazados que contienen artículos o recursos que presentan uno o más problemas. Como respuesta a un problema, se puede agregar una argumentación indicando una solución para el mismo. Dicha argumentación debe estar sustentada por una prueba o evidencia, la cual es generada por métodos de investigación; los cuales, a su vez, definen conceptos. Además, todos estos nodos y las relaciones entre ellos pueden ser asociados con críticas.

SECAL (Summarization, Evaluation, Comparison, Argumentation, and Integration), define el modelo de proceso mediante cinco pasos en la discusión. Los primeros dos

pasos (Resumen y Evaluación) se llevan a cabo en forma privada, mientras que los tres siguientes (Comparación, Argumentación e Integración) involucran a otros. Durante los dos primeros pasos, el usuario elabora una representación del recurso y lo critica, respectivamente. En el paso de la comparación, los distintos usuarios comparan los nodos y links de sus representaciones. Luego, durante la argumentación, cada uno critica las representaciones de los otros. Y por último, en la fase de integración, cada usuario une su representación con las otras para formar una representación colectiva del documento.

CLARE ha sido usado en algunos experimentos pilotos, pero los resultados han sido un poco inconclusos. Mientras que algunos usuarios sintieron que CLARE los ha ayudado a entender mejor los artículos o recursos, otros no entendían del todo como usarlos.

Chronicle: Application-Specific Versions

Un problema con los sistemas de archivos tradicionales, es que la unidad de nombrado, almacenamiento y versionado es una unidad física soportada por el sistema, en lugar de una unidad lógica específica a una aplicación. Chronicle enfoca este tema dentro del contexto de una planilla de cálculo; se desarrolló como una extensión de Lotus 1-2-3, permite que varios usuarios colaboren sobre una planilla de cálculo y compartan la información de la misma.

Chronicle soporta la asignación de un nombre a un rango, el cual es un bloque rectangular de celdas de una planilla, y permite que los usuarios decidan si los valores en un rango se almacenarán como una nueva versión. Cada versión está asociada con el nombre y valores del rango, con una anotación definida por el usuario, una identificación de usuario y una fecha.

Chronicle permite almacenar las versiones y escenarios en una base de datos compartida por usuarios distribuidos. Además, se pueden consultar las diferentes versiones de un rango específico almacenadas en la base de datos y ordenarlas por nombre de rango, nombre de usuario, y escenario. Los usuarios pueden acceder simultáneamente a la planilla, pero no a la misma versión, las modificaciones simultáneas realizadas sobre una misma parte de la planilla genera diferentes versiones que, posteriormente, pueden ser usadas como alternativas.

A.6.4 COLABORACIÓN BASADA EN SESIONES Y MENSAJES (TALK + MAIL)

Las aplicaciones colaborativas pueden combinar, al menos de dos maneras diferentes, los soportes de sesiones y de mensajes.

Como vimos en MUD, una forma es permitir que los usuarios de una sesión “murmuren” con un subconjunto de usuarios. InternetChat provee una facilidad similar, que permite que los usuarios de una sesión utilicen el correo electrónico habitual para enviar URLs e información al resto de los usuarios.

La otra forma es, como en un sistema de workflow, comenzar automáticamente una sesión colaborativa como una tarea en el workflow. Por ejemplo, en un sistema de evaluación de aspirantes, una vez recibidos por mail todos los formularios, se podría iniciar automáticamente una sesión entre los evaluadores.

A.6.5 COLABORACIÓN BASADA EN SESIONES Y OBJETOS (TALK + FILE)

En estas aplicaciones se combina la abstracción de una sesión con la noción de objeto compartido, difieren unas de otras en el objeto compartido y en el método de compartición.

RTCAL: Real-Time Artifact Sharing

RTCAL (Real-Time CALendar) fue una de las primeras aplicaciones que mostró la posibilidad de compartir objetos en tiempo real. Es un sistema de soporte de decisiones que permite a los usuarios administrar colaborativamente los horarios de las reuniones. Presenta información privada y pública: para cada usuario y horario, muestra una columna compartida indicando si el horario está libre para todos los usuarios, y otra columna privada, con los compromisos del usuario en ese horario.

Los comandos de los usuarios están divididos en comandos de aplicación y de control de conferencia. Los primeros manipulan el calendario y sólo pueden ser ejecutados por el controlador. Los segundos, son usados para pasar el control y dejar la conferencia, y pueden ser ejecutados por los participantes en cualquier momento. Existe un rol especial, *chairperson*, que tiene autoridad para finalizar la conferencia y determinar quien es el controlador actual.

Existen varias estrategias para implementar este tipo de aplicaciones. RTCAL crea una replica en la workstation de cada usuario y las mantiene consistentes. Esta estrategia es usada en casi todas las aplicaciones descritas en esta sección, con excepción de TeamWorkstation y ClearBoard, que usan un proceso centralizado, y CAIS, que combina las dos estrategias.

Cognoter: Private/Shared Windows & Process Model

Cognoter es la más antigua de las aplicaciones desarrolladas como parte del proyecto Xerox's Colab, el cual desarrolló software que puede ser usado colaborativamente por pequeños grupos (de dos a cinco integrantes) co-ubicados en un sala Colab.

Cognoter permite a los usuarios dar cuerpo a sus ideas y plasmarlas en documentos de texto estructurado. A diferencia de RTCAL, que mezcla la información privada y pública en una misma ventana, crea ventanas separadas para cada usuario: una ventana compartida de organización de ítems y otra privada de edición. La primera ventana es usada para mostrar un conjunto de ítems de texto, los cuales pueden contener otros ítems o estar asociados con anotaciones. La segunda ventana es usada para crear y modificar los ítems y sus anotaciones. Cada ítem es transferido desde la ventana de

edición privada a la ventana compartida cuando la modificación es consumada por el usuario.

Cognoter soporta tres fases físicas de colaboración: brainstorming, organización y evaluación. En la primer fase, los participantes proponen varias ideas; en la segunda, se recopilan las ideas relacionadas en posibles alternativas; y en la última fase, se eliminan aquellas que no son aceptables.

GROVE: Access-Controlled Views

Al igual que RTCAL, GROVE es un editor de grupo; pero a diferencia de éste, soporta control de acceso de granularidad fina y permite que los ítems privados sean vistos dentro del contexto de la información pública.

Un *outline* en GROVE es una estructura recursiva que esta compuesta por ítems estructurados, cada uno de los cuales está asociado con algún texto y puede contener subítems. Cada usuario despliega una vista del outline en un visualizador; dicha vista es un subconjunto del outline al que tiene acceso el usuario, y un visualizador es una ventana que muestra la vista del outline. Esta distinción permite que los usuarios compartan vistas, sin tener que compartir los visualizadores para mostrarlas.

Una vista puede ser privada, pública o compartida: la primera sólo es legible por el usuario que la creó; la segunda, por todo los usuarios; y última, sólo por un subconjunto de los usuarios.

Cuando un usuario inicia una sesión, determina de que tipo será la vista, y la misma se crea en las pantallas de todos los usuarios que la comparten. La vista muestra el contenido completo del outline que es accedido por dichos usuarios. Las anotaciones realizadas, sólo son visibles por los usuarios que comparten la vista.

Cada uno de los ítems mostrados en las vistas tiene asociado permisos de lectura, escritura y modificación, que pueden ser cambiados en cualquier momento. Un usuario con permiso de modificación sobre un ítem, puede hacer que el ítem sea de sólo lectura o escritura para un conjunto de usuarios

Los cambios realizados por un usuario sobre un ítem, son transmitidos inmediatamente al resto de los integrantes que comparten dicho ítem. Además, GROVE permite que múltiples usuarios modifiquen simultáneamente un ítem, asegurando que los cambios se visualizan en el mismo orden en que fueron realizados. Para esto, no utiliza centralización ni abortos para garantizar el orden global, sino que, de forma optimista, espera que se produzcan pocas operaciones concurrentes inconsistentes. Cuando ocurre alguna de estas operaciones, la transforma para garantizar la consistencia. Por otro lado, no soporta bloqueo, y confía en los protocolos sociales para prevenir conflictos.

CES: Tickle Locks and Version Histories

CES (Collaborative Editing System) soporta bloqueo de granularidad fina, transmisión diferida de los cambios producidos y versiones distribuidas de documentos. Permite editar colaborativamente un outline estructurado cuyos nodos hojas representan segmentos de texto de longitud variable. Cada uno de los nodos de texto pertenece a un usuario, que es considerado el autor primario del nodo, y reside en la máquina de su

propietario. Las copias secundarias del nodo son copiadas en nodos de otros autores y se mantienen consistentes con la copia primaria.

Los cambios realizados por un usuario no son transmitidos inmediatamente, sino cuando se realiza un commit. Para reducir la sobrecarga de los usuarios, CES soporta commit implícito; esto es, consume los cambios automáticamente como un efecto lateral de algunas “acciones de edición significantes”, tales como la eliminación de una palabra. De esta manera, cuando un usuario observa un nodo que está siendo modificado por otro, ve una versión desactualizada del mismo hasta que los cambios son consumados por el otro usuario.

CES garantiza que dos usuarios no pueden escribir simultáneamente sobre el mismo nodo: un usuario bloquea un nodo cuando lo edita y retiene el bloqueo mientras realice alguna actividad de edición, pero si dicho usuario permanece ocioso durante cierta cantidad de tiempo, cualquier otro usuario le puede quitar el bloqueo.

CES se implementó usando un sistema de objetos distribuido llamado Argus.

Aspects: Multiple Locking Modes

Hasta el momento vimos dos clases de bloqueo: control de granularidad gruesa en RTCAL, y control de concurrencia de granularidad fina en CES. En esta sección analizaremos una aplicación multiusuario, Aspects, que provee múltiples modelos de control de concurrencia y una interfase de usuario especial para obtener y liberar explícitamente los bloqueos.

En Aspects, una conferencia es creada por un usuario especial llamado moderador, el cual tiene derechos de acceso especiales.

Soporta tres modos de control de concurrencia o “mediación”, y el moderador es el que determina dinámicamente que modo se debe usar. En el modo “libre de competencia”, el bloqueo es a nivel de párrafo. En el modo “mediación media”, se serializa el acceso al documento completo, y los usuarios usan un mecanismo de paso de bolígrafo para pasar el control: cada usuario puede escribir, esperar el bolígrafo o simplemente observar. Y, por último, en el modo “mediación completa”, el acceso al documento también está serializado, pero el moderador es quien determina cual es el próximo usuario en tomar el bolígrafo una vez abandonado.

Al igual que Cognoter, Aspects provee soporte para la gesticulación asociando a cada usuario con un único telepointer cuya posición es visualizada por el resto de los participantes.

Groupsketch and GroupDraw: Multiple Gestures and Optimistic Locks

Hasta ahora sólo analizamos la manipulación compartida de documentos textuales. Groupsketch y Groupdraw son dos programas colaborativos desarrollados por el grupo de Saul Greenberg en Calgary, que permiten la edición grupal de dibujos. El primero, es un simple programa de dibujo que permite a los usuarios dibujar píxel por píxel; mientras que el segundo, es un programa de dibujo estructurado que permite a los usuarios crear figuras geométricas, tales como líneas o círculos, y ejecutar comandos específicos sobre ellos.

Groupsketch tiene muchas de las características de las aplicaciones que vimos anteriormente: presenta imágenes idénticas de la ventana de dibujo en las pantallas de todos los participantes, muestra los cursores de los participantes en cada una de las ventanas, permite que cualquier usuario ingrese un comentario en cualquier momento, e inmediatamente propaga las acciones a todos los usuarios.

Groupdraw es similar al anterior, pero con unas diferencias importantes, algunas de las cuales tienen que ver con que es un editor orientado a objetos: al igual que GROVE, provee control de acceso de granularidad fina, y permite a los usuarios crear objetos y determinar si serán de escritura, sólo lectura o no legible para el resto de los participantes. Provee una paleta para seleccionar y manipular objetos, no vincula las paletas de diferentes usuarios, ya que puede causar que disminuya la colaboración. Además, soporta bloqueo/desbloqueo implícito cuando un objeto es seleccionado/deseleccionado por un usuario. Cuando un usuario trata de manipular un objeto, su agente chequea contra un servidor remoto el estado de dicho objeto. Durante el tiempo que se demora en recibir la respuesta, se ejecutan las acciones sobre el objeto, y en caso que la respuesta fuera que el objeto estaba bloqueado, dichas acciones son deshechas. Esta estrategia optimista de bloqueo es similar al broadcast optimista de GROVE y funciona bien en la práctica.

Ensemble: Implicit Sessions & Multicasting Telepointers

Ensemble es otro grupo de aplicaciones de dibujo basadas en objetos, donde los usuarios no deben iniciar explícitamente una sesión, sino que simplemente usan Ensemble para acceder a los archivos gráficos. Los usuarios que acceden al mismo archivo son agregados automáticamente a la misma sesión; y no ven los cursores del resto de los participantes de la sesión, sino que exportan e importan selectivamente cursores de y para los otros usuarios.

Como en GroupKit, el bloqueo/desbloqueo tiene granularidad de objeto y se realiza implícitamente. Mientras un objeto está bloqueado, el resto de los usuarios no pueden ver los cambios producidos sobre él; sólo son visibles cuando el objeto es desbloqueado.

GroupDesign: Multiple Collaboration Modes

GroupDesign es otra herramienta colaborativa de dibujo basada en bloqueo. Cuando se inicia una operación sobre un objeto, se muestra un icono en el mismo para indicar al resto de los usuarios que el objeto está siendo manipulado. La forma del icono indica el tipo de operación que se está ejecutando y el color indica el iniciador. Cuando la operación se completa, se utiliza la animación para explicar al resto de los usuarios cual fue el resultado.

GroupDesign provee un protocolo de bloqueo más inteligente que permite a los usuarios ejecutar operaciones sobre objetos bloqueados que sean compatibles con la operación que está bloqueando el objeto. Por ejemplo, permite ejecutar simultáneamente sobre un rectángulo, las operaciones de rotación y cambio de color.

Como en GroupDraw, cada usuario puede hacer scroll a una parte diferente del dibujo,

y puede ubicar su ventana en una posición diferente. La acción de un usuario sobre un objeto se propaga a aquellos que han realizado scroll del objeto identificando el tipo de operación ejecutada.

GroupDesign provee dos modos especiales para permitir a los usuarios sincronizar las vistas a las que se les han realizado scroll a diferentes partes. En el modo *Localizador*, un usuario es “teletransportado” a las vistas de uno o más usuarios, de las cuales puede retornar a su vista. En el modo *Teleconferencia*, se vinculan las vistas de los usuarios para que observen los scrolling y cambios de la ventana de los otros usuarios. En este modo, los punteros no son compartidos, pero un objeto GroupDesign puede usarse como *telepointer*.

Provee otros dos modos útiles para soportar awareness. En el modo *Identificación*, muestra cada objeto con el color de la última persona que lo modificó. En el modo de *Edad*, usa el color para indicar la edad del objeto. Y por último, en el modo *Tiempo de Descanso*, no se ven las acciones de los usuarios hasta que se les hace un commit explícitamente.

ClearBoard: Face-to-Face Awareness

En todas las aplicaciones que analizamos hasta aquí vimos que un usuario es indirectamente consciente de las acciones de los otros a través de telepointers, representaciones de los usuarios por colores, feedback de audio, y otras características. Clearboard, que fue diseñado para la colaboración entre dos usuarios, muestra que es posible para un usuario ser directamente consciente de las acciones de un usuario remoto. Cada usuario ve al otro como detrás de una pizarra transparente, y los dibujos realizados por ambos usuarios son compartidos. Así, un usuario puede borrar o refinar el dibujo realizado por el otro.

Se utilizan espejos, proyectores y una aplicación de pizarras compartida para crear la ilusión de conciencia cara-a-cara. Las pantallas de los usuarios están inclinadas 45 grados, y la imagen de un usuario es reflejada por un espejo, capturada por una cámara en lo alto, y transmitida a un proyector en el otro sitio que muestra la imagen en la pantalla. La aplicación de pizarra compartida garantiza que los usuarios compartan los cambios de los objetos dibujados sobre la pizarra.

TeamWorkstation: Seamless Desktop/Computer Awareness

Clearboard trabaja bien siempre y cuando los únicos objetos compartidos por los usuarios sean los dibujados sobre la pizarra. En el caso que los usuarios quieran acceder a objetos físicos del escritorio distribuido y a objetos electrónicos creados por la aplicación, dichos objetos son mostrados en ventanas separadas; pero esto fuerza a los usuarios a centrarse en un solo conjunto de objetos a la vez. En lugar de esto, TeamWorkstation, crea un ambiente en el cual se muestran los dos conjuntos de objetos en una pantalla compartida simple.

TeamWorkstation utiliza la estrategia CaptureLab para compartir una cpu simple entre las pantallas local y remota: captura las señales de video enviadas por la cpu a su pantalla local, y las envía a la pantalla remota a través de la red. También envía

imágenes de cada escritorio capturadas por una cámara de video, que son combinadas con los stream generados por la cpu. Por último, soporta entrada multiusuario, captura los eventos de teclado generados por el usuario remoto y los envía al sitio local a través de la red, donde se multiplexan con los eventos de teclado del usuario local antes de ser derivados a las aplicaciones.

Digital Desk: Integrated Desktop/Computer

La necesidad de superponer las imágenes del escritorio físico con el “escritorio” computacional surge solamente cuando ambos son distintos. Este no es el caso de DigitalDesk, donde el escritorio físico sirve también como escritorio electrónico: un proyector muestra documentos electrónicos o ventanas en el escritorio físico, el cual puede ser entremezclado con papeles físicos. Los usuarios interactúan presionando y arrastrando sus dedos, lo que equivale a presionar o arrastrar un dispositivo para mover el cursor en la pantalla. Una cámara en lo alto escanea los documentos en papel y detecta los movimientos de los dedos mientras que un micrófono adherido debajo del escritorio detecta la presión de los mismos.

La aplicación DigitalDesk Calculator, muestra como esta tecnología es útil para la interacción de usuario simple. El usuario puede interactuar con la calculadora presionando sobre teclas proyectadas, o directamente pegando números desde un papel a la calculadora.

Esta tecnología es particularmente interesante desde el punto de vista multiusuario, ya que se pueden proyectar los documentos electrónicos o en papel sobre escritorios remotos, dando conciencia de ambos objetos sin la necesidad de la superposición de imágenes.

A.6.6 COLABORACIÓN BASADA EN MENSAJES Y OBJETOS (MAIL + FILE)

News Bulletin-Boards: Shared Mailboxes

Los sistemas puros basados en mensajes proveen un mecanismo para añadir mensajes a las casillas de correo privadas de los usuarios. Los sistemas puros basados en objetos permiten que múltiples usuarios compartan un objeto. New combina características de ambos tipos de sistemas: al igual que en el primero, los usuarios envían pedidos para añadir mensajes a una casilla de correo; pero de la misma forma que el segundo tipo de sistema, dicha casilla es un objeto compartido accesible por múltiples usuarios. En realidad, News Bulletin-Boards tiene que ser mucho más escalable que los otros objetos compartidos que vimos hasta aquí, debido al gran número de usuarios que lo comparten (incluyendo potencialmente a todos los usuarios de Internet) y al gran número de mensajes generados (100Mb. por día aproximadamente)

Afortunadamente, el problema de construir tales objetos escalables no es tan difícil como ocurre con las bases de datos o archivos: primero, los nuevos mensajes son inmutables. Segundo, y tal vez lo más importante, no hay necesidad de asegurar que

los usuarios ven el mismo estado del objeto compartido en cualquier momento. Dicho de otra manera, cuando los usuarios inician nuevas sesiones, no esperan recibir todos los ítems que fueron enviados antes de comenzar la sesión, sino que están contentos de recibir esos nuevos ítems en sesiones posteriores.

News utiliza una metodología simple que forma la base de Lotus Notes. Cada usuario accede al sistema usando un cliente News, el cual se conecta con un servidor News que mantiene un repositorio con todos los ítems. El cliente News descarga dichos ítems y envía al servidor los nuevos ítems agregados por su usuario. Existen varios servidores replicados en Internet, conectados directa o indirectamente con otros servidores, que mantienen sus repositorios consistentes propagando los ítems que reciben a los servidores que están conectados directamente. Esta metodología produce que los servidores reciban ítems duplicados a través de diferentes caminos, pero la duplicidad es fácilmente reconocible ya que cada ítem publicado tiene un único identificador.

Los servidores pueden eliminar ítems para liberar espacio en disco, lo que interfiere con la garantía de que un ítem debe ser recibido por todos los usuarios.

Message Filtering

Aún cuando el sistema pueda manejar grandes cantidades de tráfico, resulta complicado para un usuario ordenar los ítems recibidos para encontrar información de su interés. La solución a este problema es usar técnicas de filtrado para categorizar los mensajes de acuerdo a diferentes criterios. Dicha categorización puede ser usada directamente por el usuario como una característica del producto o automáticamente por “agentes” para eliminar ciertos mensajes.

Se pueden usar diversos criterios de para clasificar un mensaje:

- Grupo de noticias: el criterio más común para seleccionar nuevos mensajes es por el grupo de noticias al cual pertenece.
- Hilo de discusión: discusión en la cual aparece el mensaje; es una característica soportada por News y Notes Mail, en la cual se vincula un mensaje con una discusión.
- Respuestas Urgentes: si es una respuesta a un mensaje enviado por el receptor. Los usuarios están más interesados en mensajes que son parte de conversaciones urgentes en la cuales están involucrados.
- Emisor/Asunto/Contenido: palabras o frases que aparecen en los campos emisor, asunto y contenido de los nuevos ítems. Algunos clientes News soportan “kil files” que identifican las palabras o textos que no son de interés para el usuario y elimina el mensaje seleccionado. Por el contrario, otros permiten a los usuarios especificar palabras o frases para seleccionar los mensajes que desean encontrar.
- Esfuerzo del emisor: cuanto esfuerzo es requerido por el emisor para enviar un mensaje: se puede distinguir entre enviar un mail a una lista de personas, o enviarlo individualmente.
- Moderador seleccionado: Whether some moderator has selected it, a feature supported in moderated news groups
- Clasificación: las clasificaciones dadas por usuarios arbitrarios en el sitio local,

como vimos en el sistema Tapestry. Esta es la generalización de la idea del moderador, que permite a múltiples usuarios opinar sobre un mensaje,

GroupLens: Extensible Aggregation-based Filtering

En comparación con la estrategia basada en un moderador, Tapestry es más sofisticado en considerar la entrada de múltiples usuarios; pero es más difícil de usar y menos eficiente ya que el usuario debe considerar todas las categorizaciones y el sistema debe almacenarlas con el mensaje. Posiblemente, esta sea la razón por la cual Tapestry restringe las clasificaciones a un sitio simple. GroupLens muestra que es posible alcanzar los beneficios de ambas estrategias.

Al igual que Tapestry, permite que múltiples usuarios clasifiquen un artículo; pero a diferencia de éste, permite que dichos usuarios estén en sitios arbitrarios y, para cada nuevo lector del artículo, deduce como el nuevo lector deseará ver el mensaje.

La idea de presuponer las preferencias del nuevo lector es simple e intuitiva: se puede deducir cómo le gustaría ver a un lector un mensaje, analizando los gustos de otros lectores con similares preferencias; y se pueden determinar cuán similares son las preferencias de dos lectores por correlación en las clasificaciones que ambos han realizado sobre los mismos mensajes.

GroupLens implementa esta idea, de manera extensiva, sobre la arquitectura de News. Agrega a los clientes y servidores un proceso llamado “better bit bureau”, el cual es responsable de implementar la semántica adicional. Un cliente News modificado presenta nuevas clasificaciones al usuario y también recolecta las de él.

POLITeam: Interleaved Message- and Artifact- based Collaboration

Algunas veces la colaboración basada en mensajes y la basada en objetos son dos modos de colaboración para lograr una misma tarea. Consideremos el desarrollo compartido de un documento: los usuarios pueden enviarse unos a otros el objeto por correo electrónico, o crearlo como un objeto compartido. En estas situaciones, es útil integrar las dos alternativas permitiéndoles a los usuarios cambiar fácilmente entre ellas.

POLITeam fue diseñado exactamente para soportar el desarrollo del discurso de un ministro. Varias organizaciones están involucradas en la composición y comentario de un discurso; tradicionalmente esta tarea de colaboración se lograba derivando el documento a través de todas las organizaciones pertinentes. POLITeam automatiza y distribuye el documento proporcionando un sistema de workflow que automáticamente dirige el documento a su próximo destino una vez que el usuario termina de procesarlo. En el método tradicional, se serializa el acceso al documento hasta que el usuario termina de trabajar con el mismo, nadie más puede accederlo. POLITeam extiende este esquema permitiendo al dueño actual del documento compartirlo temporalmente con otros usuarios (como el destinatario anterior), reduciendo de esta manera la sobrecarga de enviar el documento continuamente de un lado a otro entre los distintos usuarios. De esta manera, permite intercalar la colaboración basada en mensajes con la basada en objetos durante el procesamiento de un documento, y, consecuentemente, brinda

otra manera de integrar estos dos tipos de colaboración.

Puede pensarse de un híbrido entre el correo y las noticias, que permiten cambiar dinámicamente entre sus características: los primeros soportan mensajes privados en buzones mientras que los últimos soportan artículos de noticias públicas en pizarras de anuncios. POLITeam provee un esquema para transformar dinámicamente un objeto de un mensaje privado a un objeto compartido, y viceversa.

Linking Artifacts and Messages

Otra manera de integrar la colaboración basada en mensajes y la basada en objetos es conectar mensajes privados con objetos públicos. Algunos sistemas de mensajería conectan objetos a mensajes, en los cuales se conocen los nombres de dichos objetos (tales como URLs), y permiten a los usuarios presionar sobre los objetos para abrir el visualizador del objeto correspondiente. Al contrario de Quilt, donde los mensajes sobre objetos pueden ser conectados directamente a ellos. Además, en Quilt, los mensajes sobre las actividades de un usuario sobre un objeto, pueden ser enviados automáticamente por el sistema a los colaboradores que han expresado interés en ellos.

Mercury: Automatic Error Notification

Mercury también genera, para los colaboradores, notificaciones basadas en objetos; pero las muestra dentro del contexto del objeto y determina automáticamente, basándose en los objetos que están manipulando los usuarios, cuales de ellos recibirán las notificaciones. Los objetos son programas formados por módulos distribuidos, cada uno de los cuales tiene un único programador responsable. El sistema permite que diferentes programadores vean o modifiquen diferentes programas simultáneamente, pero un programa sólo puede ser consultado o modificado por su programador responsable.

Mercury deriva de la tecnología de editores basados en el lenguaje. Tales editores entienden la estructura, semántica y sintaxis de los programas, e informan progresivamente al usuario sobre errores de semántica y sintaxis. Mercury extiende esta notación informando progresivamente a los programadores responsables sobre errores en las interfases de los módulos que están editando. Cuando un programador realiza un cambio en una interfase exportada por un módulo, y causa inconsistencias en otro módulo que está siendo consultado por otro programador, Mercury informa al último sobre los errores ocurridos. También provee escenarios que permiten ver a los programadores si los cambios realizados por ellos causan problemas en otros módulos. Mercury coloca un “firewall” sobre el módulo que está en estado inconsistente y no chequea su consistencia con los otros módulos hasta que su propietario realice un commit. Como resultado de esto, no se generan mensajes falsos en el caso que dos programadores realicen simultáneamente cambios que son inconsistentes individualmente, pero no colectivamente con el programa original. Si un cambio en una interfase produce inconsistencias en un módulo, cuyo propietario no está conectado o

no está consultando el módulo, Mercury guarda la información y muestra el mensaje de error la primera vez que el módulo sea consultado por su propietario.

A.6.7 COLABORACIÓN BASADA EN SESIONES, MENSAJES Y OBJETOS (FILE + MAIL + TALK)

Astro-VR: MUD++

Astro-VR es una extensión de MUD, provee a los astrónomos un ambiente donde pueden reunirse y dar presentaciones breves. Dentro de las facilidades singulares que presenta se incluyen: un correo electrónico “auto-controlado” y un boletín informativo restringido a los participantes, links compartidos y definidos por el usuario hacia imágenes astronómicas, un editor/visor para las presentaciones de texto e imágenes en salas especiales de conferencia que mantienen el hilo de las discusiones, y acceso colaborativo a programas existentes usados por astrónomos.

Forum: Distributed Presentations

Forum es una herramienta que fue diseñada para permitir a los interlocutores dar una presentación ante una gran audiencia distribuida. Cada miembro de la audiencia ve las transparencias y un video del interlocutor; además, puede presionar un botón para levantar la mano cuando desee realizar una consulta, y el interlocutor ve los nombres de los participantes que levantaron la mano. El conferencista puede invitar a uno de los usuarios para que diserte, entonces se muestra una imagen guardada de dicho usuario al conferencista y al resto de los oyentes. Los estudiantes pueden votar en respuesta a una encuesta, y los resultados son mostrados a todos los usuarios. También pueden enviar mails privados al interlocutor o otro estudiante.

IRI: Video Windows and Application Sharing

IRI (Interactive Remote Instruction) es similar al anterior en que permite a los miembros de una audiencia observar al interlocutor y a las transparencias, y enviar mails anónimos al conferencista en respuesta a encuestas. Pero en lugar de mostrar imágenes estáticas de los usuarios, muestra videos de los estudiantes que han levantado su mano. Al igual que Astro-VR, permite a los usuarios compartir una aplicación existente, tal como AutoCad.

CAIS: Interleaving Session- and Non Session-based Collaboration

En todas las herramientas que vimos hasta aquí, los usuarios manipulan un objeto en forma privada o en una sesión compartida. CAIS (Collaborative Asynchronous Inspection System) muestra los beneficios de mezclar estos dos modos de colaboración. Las inspecciones tradicionales suponen que la inspección completa se realiza durante la reunión. En cambio, CAIS cambia este paradigma para permitir que parte de la tarea se realice asincrónicamente, y así reducir la sobrecarga de la administración de las reuniones. La aplicación soporta hilos de discusión, votaciones y resúmenes de faltas.

Cuando un usuario concluye una sesión asincrónica individual, automáticamente se envían mails al resto de los participantes con respecto a los comentarios y votos ingresados por el usuario.

Al igual que CES, CAIS distribuye la información compartida entre objetos autónomos distribuidos. En particular, crea un objeto distribuido separado para cada comentario, documento que se visualiza, log de historial, jerarquía de hilos de discusión, lista de faltas y resumen de inspección.

VisTool: Editable, Piped, Collaborative Views

VisTool es una aplicación para automatizar el proceso de administración de los interlocutores externos a un departamento. Asocia a cada visitante con información sobre el título, abstract, horario y sala de la presentación, y a la administración de la reunión con miembros del departamento. Los usuarios pueden crear vistas de la base de datos ingresando consultas o patrones que especifican la información de interés. Al igual que en las bases de datos, estas vistas pueden ser en cascada, es decir, que pueden ser creadas a partir de otras vistas. Pero a diferencia de las vistas de las bases de datos, son editables en modo de actualización y los cambios realizados sobre ellas son reflejados en las vistas/base de datos de las cuales derivan. Estas actualizaciones también provocan que se modifiquen los objetos relacionados. En particular, cuando se agenda una reunión entre un visitante y los miembros del departamento, automáticamente se realizan las actualizaciones correspondientes en las agendas de los últimos. Más aún, cuando los usuarios modifican concurrentemente el mismo objeto, son puestos en una misma sesión en la que pueden detectar conflictos y negociar.

Además de vistas editables, VisTools provee comandos shell para modificar y visualizar la base de datos y sus vistas. Un usuario también puede asociar operaciones de vistas definidas por un programador, tales como AgregarCompromiso o EliminarCompromiso, con manejadores que se invocan automáticamente cuando se ejecuta la operación. Por lo general, estos manejadores usan el mail para enviar mensajes a los usuarios y toman como argumentos los parámetros de la operación. Así, los usuarios finales pueden incrementar la funcionalidad provista por los programadores de la aplicación.

BIBLIOGRAFIA

Ellis, C., Gibbs, S., and Rein, G., Groupware: Some Issues and Experiences, Communications of the ACM, Vol. 34, No. 1, 1991, 39-58

Prasun Dewan, Rajiv Choudhary, and Honghai Shen. An Editing-Based Characterization of the Design Space of Collaborative Applications. Journal of Organizational Computing, 1994.

David Coleman. Groupware - Collaborative Strategies for Corporate LANs and Intranets. Prentice-Hall, 1997. ISBN: 0-13-727728-8.

Saul Greenberg. Computer-supported Cooperative Work and Groupware. Academic Press, London, 1991.

Saul Greenberg. Personalizable Groupware: Accommodating individual roles and group differences, 1991.

Grudin, J., 1994. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37, 1, 92-105.

Grudin, J. and Poltrock, S. (1991). *Computer-supported cooperative work*. CHI'91 Tutorial notes.

Volker Wulf. Negotiability: A Metafunction to Tailor Access to Data in Groupware, 1995.

The Yankee Group. Communication, Collaboration, Coordination: The "Three Cs" of Workgroup Computing. www.yankeegroup.com, 2001.

P. Brusilovsky: "*Adaptive hypermedia*". User Modeling and User Adapted Interaction, Ten Year. Anniversary Issue, (Alfred Kobsa, ed.) 11 (1/2), 87-110. 2001
15-17 de Noviembre de 2000 Puertollano (Ciudad Real)

M. Constantino-Gonzalez, D. Suthers: A Coached Collaborative Learning Environment for Entity-Relationship Modeling In *Intelligent Tutoring Systems*, Proceedings of the 5th International Conference (ITS 2000).

- J. Ewing, D. Miller: "A framework for evaluating computer supported collaborative learning". Educational Technology & Society 5(1) 2002. ISSN 1436-4522
- S. Gerry (Editor 2002) "Proceedings of CSCL 2002, Boulder, Colorado, USA". January 7 - 11, 2002, distributed by Lawrence Erlbaum Associates, Inc. Hillsdale, New Jersey, USA © 2002
- C. R. Haller, V. J. Gallagher, L. T. Weldon, R. M. Felder, "Dynamics of peer education in cooperative learning workgroups." J. Engr. Education, 89(3), 285-293 (2000).
- L. Lipponen: "Exploring foundations for computer-supported collaborative learning". Proceedings of CSCL 2002, Boulder, Colorado, USA. January 7 - 11, 2002 Edited by Gerry Stahl, distributed by Lawrence Erlbaum Associates, Inc.
- C. J. M, Olguín, A. L. N. Delgado, I. L. M. Ricarte: "An Agent infrastructure to set collaborative environments". Educational Technology & Society 3(3) 2000. ISSN 1436-4522
- R. Oppermann, R. Rashev, Kinshuk: "Adaptability and Adaptivity in Learning Systems". Knowledge Transfer (volume II) (Ed. A. Behrooz), pAce, London, pp173-179. 1997. (ISBN 1-900427-015-X).
- A. Rachida, M. Amine Benkiran, M. Ato.: "A Framework for Adaptive and Cooperative Learning for the Internet; SMART Learning". INET 2000 Proceedings. Japan. 18-21 July 2000
- K. Sinitsa. "Learning Individually: a Life-Long Perspective Introduction to the Special Issue". Educational Technology & Society 3(1)2000. SIN 1436-4522
- Kahler, H., A. Mørch, O. Stiemerling and V. Wulf (2000), "Computer Supported Cooperative Work: The Journal of Collaborative Computing". In: Computer Supported Cooperative Work: The Journal of Collaborative Computing, Volume 9, No 1. 2000, Kluwer Academic, pp. 1-4, ISSN 0925-9724.
- Mackay, W.E. (1990). "Patterns of sharing customisable software". CSCW'90 In Proceedings of the conference on computer supported co-operative work, 7-10 October (ACM Press, LA), p.209-221.
- Heeren, E.(Ed.), M. Biemans, H. ter Hofte, I. Mulder, J. Schuurman, R. Slagter and M. Veenstra (1999). "Open Computer-Supported Collaborative Work: Outline of research". Telematica Instituut, Enschede, TI Reports series (TI/RS/99017). <https://extranet.telin.nl/docuserver/dscgi/ds.py/Get/File-847>.

Hettinga, M. and E. Heeren (1998), "Evolution: The undervalued source of design guidelines for cooperative technology". In: Darses, F., & Zaraté, P. (Eds.), *Proceedings of COOP'98: Volume II* (pp. 29-31), Cannes, France: INRIA.

Koch, M. and G. Teege (1999), "Support for tailoring CSCW systems: adaptation by composition". In: *Proceedings of 7th Euromic workshop on parallel and distributed processing*, Funchal, Portugal, IEEE Press, pp. 146-152.

Stiemerling, O. and A.B. Cremers (1998), "Tailorable component architectures for CSCW-systems". In *Proceedings of the 6th Euromicro workshop on parallel and distributed programming*, January 21-24 1998, Madrid, Spain, IEEE Press, p. 302-308.

Mørch, A.I. (1997), "Three Levels of End-User Tailoring: Customization, Integration, and Extension". In: *Computers and design in context*. M. Kyng & L. Mathiassen (eds.). The MIT Press, Cambridge, MA, 1997, pp. 51-76.

MacLean, A., K. Carter, L. Lovstrand and T. Moran. (1990), "*User-tailorable systems: Pressing the issues with buttons*". In: J. Whiteside (eds), *Human Factors in computing systems, Proceedings of the conference on Computer Human Interaction*, ACM Press, pp. 175-182.

Oppermann, R. and H. Simm (1994), "Adaptability: user-initiated individualisation". In Opperman, R. (Ed). *Adaptive user support, Ergonomic design of manually and automatically adaptable software*. Lawrence Erlbaum, Hillsdale.

V. Bellotti, S. Bly (1996): *Walking away from the desktop computer: distributed collaboration and mobility in a product design team*. In: *Proceedings of the ACM 1996 conference on Computer-Supported Cooperative Work (CSCW'96)*, 1996, S. 209 - 218.